7-2021

Phase-Field Modeling of the Polymer Membrane Formation Process for Micro- and Ultra-Filtration

Michael Rosario Cervellere
University of Arkansas, Fayetteville

Phase-Field Modeling of the Polymer Membrane Formation Process
for Micro- and Ultra-Filtration

A dissertation submitted in partial fullfillment
of the requirements for the degree of
Doctor of Philosophy in Engineering
with a concentration in Mechanical Engineering

by

Michael Rosario Cervellere
University of Arkansas
Bachelors of Science in Mechanical Engineering, 2016

July 2021
University of Arkansas

This dissertation is approved for recommendation to the Graduate Council.

Paul C. Millett, PhD
Dissertation Director

David M. Ford, PhD
Committee Member

Han Hu, PhD
Committee Member

Arun Nair, PhD
Committee Member

Xianghong Qian, PhD
Committee Member

Abstract

Porous polymer membrane filters are widely used in separation and filtration process. Micro- and ultra-filtration membranes are commonly used in biopharmaceutical applications such as filtering viruses and separating proteins from a carrier solution. The formation of these membrane filters via phase inversion is a complex and interconnected process where varying casting conditions can have a wide variety of effects on the final membrane morphology. Tailoring membrane filters for specific performance characteristics is a tedious and time consuming process. The time and length scales of membrane formation make it extremely difficult to experimentally observe membrane formation. Modeling the membrane formation process allows one to slow down time and closely observe the formation of complex pore networks. This allows new understanding and visual representations of the effects of different casting conditions and the resulting pore networks that form.

This dissertation presents two separate models for two different membrane formation processes - thermally induced phase separation (TIPS) and non-solvent induced phase separation (NIPS). The Phase-Field method is employed to model the mesoscopic morphological structures that emerge during membrane formation. The Cahn-Hilliard equation is used to capture the diffusive nature of membrane formation and the Flory-Huggins free energy of mixing describes the thermodynamic equilibriums of the polymer solutions. Large-scale two- and three-dimensional simulations are used for capturing the resulting pore morphology under various casting conditions.

The model for TIPS evaluated different casting conditions for the membrane forming system PVDF/DPC (polyvinylidene fluoride/diphenyl carbonate). The effects of casting surface temperature, thermal conductivity, quench rate, and polymer concentration were investigated. Isotropic thermal quenches were used to analyzed pore size and interconnectivity while varying the quench rate and polymer concentration. Anisotropic thermal quenches in which a constant temperature cooling surface is imposed at the top surface were used to an-

alyze the pore size and formation of a dense pore region near the casting surface. Different polymer concentrations, casting surface temperatures, and polymer thermal conductivities were used to characterize their effects on pore morphology. The formation of a dense pore region occurs near the top surface and the thickness of this dense pore region is dependent on the previous mentioned casting conditions.

The model for NIPS evaluated the effects of coagulation bath composition and polymer concentration on final membrane morphology for PES/NMP/Water (polyethersulfone/n-methyl-2-pyrrolidone/water). These simulations were then compared to handcast membranes of a similar membrane forming system for comparison the simulated structures. The NIPS model evaluated coagulation bath composition and polymer concentration and the resulting morphologies. Two and three dimensional simulations were used to look at cross section and top surface morphology respectively. The concentration of NMP in the coagulation bath was varied to verify the predictive capabilities of the model. The resulting simulations of cross sections and top surfaces had good agreement with handcast membranes when NMP is added to the coagulation bath.

Acknowledgements

I would like to give thanks to the many parties involved with my education and research. First I would like to thank my primary advisor, Paul C. Millett. He is a kind and patient man who's brilliance, organization, and leadership served as a benchmark that I'll strive to maintain throughout my professional and personal life. Second I would like to thank the MAST (Membrane Science, Engineering, and Technology) Center and the Industry Advisory Board for continued guidance during my research. Third I would like to thank the NSF INTERN program and the host organization MilliporeSigma for creating an opportunity for me to experience research and development in the private sector. Specifically from MilliporeSigma I would like to thank my primary mentors Christina Carbrello and Sal Giglia, for their high standards and continuous support gave me continued confidence to tackle the seemingly impossible. Last - but not least - I would like to thank my loving wife Lisa Rebello who I met during my first year as PhD student. Her never ending support and patience allowed me to fully embrace the challenges that I overcame during graduate school.

Contents

Appendix

List of Published Papers

CHAPTER 4: M. Rosario Cervellere, Yuan-Hui Tang, Xianghong Qian, David M. Ford, and Paul C. Millett. Mesoscopic simulations of thermally-induced phase separation in PVDF/DPC solutions. *Journal of Membrane Science.* 266-273, 2019.

CHAPTER 5: M. Rosario Cervellere, Xianghong Qian, David M. Ford, Christina Carbrello, Sal Giglia, and Paul C. Millett. Phase-field modeling of non-solvent induced phase separation (NIPS) for PES/NMP/Water with comparison to experiments. *Journal of Membrane Science.* 118799, 2021.

Chapter 1

Introduction

1.1 Motivation

The membrane formation process typically involves casting a polymer solution that is subject to non-equilibrium conditions causing phase separation [1–4]. During this process the polymer precipitates out of solution in a typical spinodal decomposition and the resulting structures result in the membrane. Performance characteristics fo these membranes are tailored for performance and selectivity, however this process can be tedious and time consuming. There are many process variables to control during membrane formation and the effects of each variable are co-related. There are many ways to create a porous polymer membrane including thermal induced phase separation (TIPS), non-solvent induced phase separation (NIPS), vapor induced phase separation (VIPS), and a combination of these methods. The work in this dissertation focuses on modeling the membrane formation process via TIPS and NIPS.

There are many process parameters that need to be considered when casting membrane filters. The recipe for polymer solution is a main consideration and can have vast affects on final membrane morphology and performance with all other process conditions held constant. The amount of polymer in solution determines the resulting morphology and increasing the polymer solution changes the resulting morphology from polymer droplets, a bicontinuous structure, and finally a cellular structure [2]. Additives into the polymer solution can be used to drastically change the way the solution reacts during phase separation/non-equilibrium conditions [5]. Additives can be used to increase solution viscosity and change the phase diagram and mode of phase separation which result in a change in morphology [6–10]. For the NIPS and TIPS solvents and salts can be added to the coagulation bath to achieve a desired morphology [11–14]. For TIPS and NIPS the temperature of the casting surface may be adjusted as well as the coagulation bath temperature with increased temperatures yielding larger pore sizes and higher throughput [15–19]. More recent advancements in NIPS include co-casting multiple layers in order to achieve a desired performance characteristic.

Experimental research in membrane formation has led to many significant advancements and marketable products. The research and development process for developing new membranes and/or adapting membrane performance for new applications is a tedious and intensive process. The initial exploratory phase of developing a new membrane can take inordinate amounts of time to find process windows that will produce membranes with desired characteristics. The time and length scales involved in the creation of membrane filters are in the scale of milliseconds to seconds and micro- to nano-meters. These small temporal and spatial attributes of membrane formation create a unique challenge for experimentalists to fully understand the underlying mechanisms occurring and the process in which certain morphologies arise given a set of process conditions.

Computational tools are becoming an evermore powerful tool in understanding the behavior of materials in various time and length scales. Computational researchers have chipped away at understanding the membrane formation process. Early efforts in simulating the membrane formation process were limited by the computational power of the machines they were using at the time. These efforts were usually oversimplifications of the physical processes - either dimensionally or representationally [20, 21]. Though these efforts were limited in complexity they still aided in the fundamental understanding into the processes in which membrane characteristics could be tailored. As the field of computational research grew so did the size of the problems they were able to tackle. The development of faster computers and more advanced modeling techniques has accelerated the field in the past decade and modern computational research takes advantage of highly parallel computational architectures resulting in large three dimensional representations/domains.

Computational work does have its limitations. Computational work can be used as a guide to experimental work, however, simulations still need experimental verification. Computational work also suffers from assumptions in order to simulate larger domains. A DPD model could theoretically simulate the thickness of an entire membrane, however, doing so would come at a high computational cost and a simulation time that would be unfeasible. In order to bridge the gap from molecular interactions to polymer chain behavior to phase behavior assumptions must be made at every level. A truly comprehensive computational model for the membrane formation process would tie in all of these scales to fully understand and accurately simulate complicated membrane structures created with complex recipes and process conditions. Until all of these methods are combined - a great problem to attempt - the simulation methods presented here and elsewhere will always fall short of completely accurate representations of experimental results.

1.2 Dissertation Objectives

The main objective of this dissertation is to model the membrane formation process of micro- and ultra-filtration via thermally induced phase separation (TIPS) and non-solvent induced phase separation (NIPS). The models utilize the phase field method for capturing the formation of pores in large two and three dimensional domains. Post processing algorithms have been used to determine pore size and interconnectivity. The following is the list of objectives for this Dissertation:

1. Create a computational model for thermally induced phase separation (TIPS).
2. Study the morphological differences of isotropic thermal quenches by varying quench rate and polymer concentration.
3. Analyze the pore size for different isothermal quench rates and polymer volume fractions.
4. Analyze the effect of increasing polymer concentration and the resulting transition from bicontonuous to cellular morphology for different quench rates.
5. Perform anisotropic thermal quenches of three-dimensional domains to understand the effect of different quenching surface temperatures.

6. Analyze anisotropic thermal quenches and quantify the thickness of the dense pore region in response to different quenching temperatures and polymer concentrations.

7. Create a computational model for non-solvent induced phase separation (NIPS) and compare model with experimental results.

8. Perform two-dimensional simulations of NIPS model to analyze anisotropic pore structures and how adding solvent to the coagulation bath changes morphology.

9. Perform three-dimensional simulations to study surface pore size and how adding solvent to the coagulation bath changes the surface morphology.

10. Handcast membranes with similar coagulation bath compositons as the NIPS model to compare cross section morphology with objective 8 and top surface morphology with objective 9.

1.3 Dissertation Structure

This dissertation is composed of six chapters with the post processing algorithms used to analyze the simulations presented in the Appendix. This chapter, the introduction, outlines the motivation behind this work and the dissertation objectives. This dissertation contains two articles both published in the *Journal of Membrane Science*. The article in chapter 4 that addresses objectives 1 - 6 is titled *"Mesoscopic simulations of thermally-induced phase separation in PVDF/DPC solutions"*. The article in chapter 5 that addresses objectives 7 - 10 is titled *"Phase-field modeling of non-solvent induced phase separation (NIPS) for PES/NMP/Water with comparison to experiments"*.

Given the brief overview of literature and methods used in the two articles in chapters 4 and 5, deeper explanations of these topics will be included in chapters 2 and 3. Chapter 2 includes a more detailed literature review of phase inversion and the previous modeling efforts for simulating membrane formation. Chapter 3 includes a more thorough description of the governing equations, computational methods, and analysis used for modeling and analyzing the TIPS and NIPS models. Chapter 3 also includes a brief explanation of the experimental method used for handcasting membranes for comparison to the NIPS model. Chapter 6 contains the conclusions of this dissertation.

Bibliography

[1] P. van de Witte, P.J. Dijkstra, J.W.A. van den Berg, and J. Feijen. Phase separation processes in polymer solutions in relation to membrane formation. *Journal of Membrane Science*, 117(1):1 – 31, 1996.

[2] Gregory R. Guillen, Yinjin Pan, Minghua Li, and Eric M. V. Hoek. Preparation and characterization of membranes formed by nonsolvent induced phase separation: A review. *Industrial & Engineering Chemistry Research*, 50:3798–3817, 2011.

[3] Agnieszka K. Holda and Ivo F.J. Vankelecom. Understanding and guiding the phase inversion process for synthesis of solvent resistant nanofiltration membranes. *Journal of Applied Polymer Science*, 132:42130, 2015.

[4] Jeong F. Kim, Ji Hoon Kim, Young Moo Lee, and Enrico Drioli. Thermally induced phase separation and electrospinning methods for emerging membrane applications: A review. *AIChE Journal*, 62:461–490, 2016.

[5] Mohtada Sadrzadeh and Subir Bhattacharjee. Rational design of phase inversion membranes by tailoring thermodynamics and kinetics of casting solution using polymer additives. *Journal of Membrane Science*, 441:31–44, 2013.

[6] Jeong-Hoon Kim and Kew-Ho Lee. Effect of peg additive on membrane formation by phase inversion. *Journal of Membrane Science*, 138:153–163, 1998.

[7] Hong-Bin Li, Wen-Ying Shi, Yu-Feng Zhang, Dong-Qing Liu, and Xiao-Feng Liu. Effects of additives on the morphology and performance of ppta/pvdf in situ blend uf membrane. *Polymers*, 6:1846–1861, 2014.

[8] Ehsan Saljoughi, Mohammad Amirilargani, and Toraj Mohammadi. Effect of peg additive and coagulation bath temperature on the morphology, permeability and thermal/-chemical stability of asymmetric ca membranes. *Desalination*, 262:72–78, 2010.

[9] Enrica Fontananova, Johannes C. Jansen, Alessandra Cristiano, Efrem Curico, and Enrico Drioli. Effect of additives in the casting solution on the formation of pvdf membranes. *Desalination*, 192:190–197, 2006.

[10] Seong Hyun Yoo, Jong Hak Kim, Jae Young Jho, Jongok Won, and Yong Soo Kang. Influence of the addition of pvp on the morphology of asymmetric polyimide phase inversion membranes: efect of pvp molecular weight. *Journal of Membrane Science*, 236:203–207, 2004.

[11] Jai Xu, Yuanyuan Tang, Yuhong Wang, Baotian Shan, Liangmin Yu, and Congjie Gao. Effect of coagulation bath conditions on the morphology and performance of psf membrane blended with a capsaicin-mimic copolymer. *Journal of Membrane Science*, 455:121–130, 2014.

[12] M. Khayet, C.Y. Feng, K.C. .Khulbe, and T. Matsuura. Study on the effect of a non-solvent additive on the morphology and performance of ultrafiltration hollow-fiber membranes. *Desalination*, 148:321–327, 2002.

[13] Yang Zhang, Lin Ye, Bopeng Zhang, Yongsheng Chen, Weigao Zhao, Guang Yang, Jie Wang, and Hongwei Zhang. Characteristics and performance of pvdf membranes prepared by using nacl coagulation bath: Relationship between membrane polymorphous structure and organic fouling. *Journal of Membrane Science*, 579:22–32, 2019.

[14] Siti Salwa Alias, Zawati Harun, and Muhamad Fikri Sohur. Effect of monovalent and divalent ions in non-solvent coagulation bath-induced phase inversion on the characterization of a porous polysulfone membrane. *Polymer Bulletin*, 76:5957–5979, 2019.

[15] M. L. Yeow, Y. T. Liu, and K. Li. Morphological study of poly(vinylidene fluoride) asymmetric membranes: Effect of the solvent, additive, and dope temperature. *Journal of Applied Polymer Science*, 92:1782–1789, 2004.

[16] Jing-Feng Li, Zhen-Lian Xu, and Hu Yang. Microporous polyethersulfone membranes prepared under the combined precipitation conditions with non-solvent additives. *Polymers for Advanced Technologies*, 19:251–257, 2007.

[17] Xuyun Wang, Lin Zhang, Dahai Sun, Quanfu An, and Huanlin Chen. Effect of coagulation bath temperature on formation mechanism of poly(vinylidene fluroide) membrane. *Journal of Applied Polymer Science*, 110:1656–1663, 2008.

[18] Ali Akbari and Reza Yegani. Study on the impact of polymer concentration and coagulation bath temperature on the porosity of polyethylene membranes fabricated via tips method. *Journal of Membrane and Separation Technology*, 1:100–107, 2012.

[19] Jinming Peng, Yanlei Su, Wenjuan Chen, Qing Shi, and Zhongyi Jiang. Effect of coagulation bath temperature on the separation performance and antifouling property of poly(ether sulfone) ultrafiltration membranes. *Industrial & Engineering Chemistry Research*, 49:4858–4864, 2010.

[20] B.F Barton and A.J McHugh. Modeling the dynamics of membrane structure formation in quenched polymer solutions. *Journal of Membrane Science*, 166:119 – 125, 2000.

[21] Dongmei Li, William B. Krantz, Alan R. Greenberg, and Robert L. Sani. Membrane formation via thermally induced phase separation (tips): Model development and validation. *Journal of Membrane Science*, 279:50 – 60, 2006.

‘

Chapter 2

Scientific Background

2.1 Phase Inversion for Membrane Formation

Polymeric membrane filters have been developed for a wide range of applications including gas separation, reverse osmosis, micro- and ultra-filtration [1, 2]. The specific application of the membrane filter determines the target morphology and pore structures. Micro- and ultra-filtration typically require highly porous membranes and pore size suitable for size exclusion of the filtration target. Membrane filters can be prepared in a variety of ways, however, phase inversion is a typical and reliable method for fabrication. Phase inversion is a technique in which a thermodynamically stable polymer solution is subject to conditions in which solubility becomes unfavorable and the polymer precipitates out of solution forming a membrane filter [1, 3]. The polymer rich regions solidify and form the membrane structure while the polymer poor regions (solvent) are subsequently removed from the membrane structure. This can be achieved by several different processes including vapor induced phase separation (VIPS), thermally induced phase separation (TIPS), and non-solvent induced phase separation (NIPS). VIPS involves bringing a polymer solution in contact with a vapor phase and as the vapor diffuses into the polymer solution the polymer precipitates out of solution and forms a membrane [4]. Due to the slower process in which phase inversion occurs during VIPS it can be easier to control the morphology with the caveat of taking a much longer time as compared to TIPS and NIPS. During the TIPS process a thermodynamically stable polymer solution is brought into contact with a temperature controlled surface and the polymer precipitates out of solution [5]. During the NIPS process a thermodynamically stable polymer solution is brought into contact with a coagulation bath which allows the non-solvent of the bath and the solvent in the polymer solution to inter-diffuse and as non-solvent replaces solvent in the polymer solution the polymer precipitates out of solution [6]. The focus of this Dissertation will be primarily membrane formation via TIPS and NIPS.

2.1.1 Thermally Induced Phase Separation (TIPS)

Thermally induced phase separation (TIPS) first introduced in the 1980s-90s is a common method for producing membranes due to its simplicity, reproducibility, and low occurrence of defects [7–16]. TIPS typically occurs when a poor solvent and polymer are heated until the polymer dissolves into solution. Subsequent quenching of the solution causes the polymer to precipitate out of solution resulting in a porous membrane [7]. After polymer precipitation and vitrification the solvent is removed from the membrane structure [17].

The TIPS process is highly dependent on polymer concentration and a schematic is shown in figure 2.1 showing the different morphological structures resulting from low to high polymer concentrations. Regardless of polymer concentration TIPS generally uses the same method for membrane formation. The homogeneous polymer solution initially starts at an elevated temperature above the critical point and then the temperature is then lowered until liquid-liquid demixing occurs resulting in phase separation into polymer rich and polymer poor regions. When the polymer concentration is sufficiently high the solution does not undergo liquid-liquid demixing and instead a spherulitic morphology emerges as solid-liquid separation occurs. To enter directly into the spinodal region upon quenching the polymer concentration must be the same as the critical point where the spinodal and binodal lines meet, however, this is not absolutely necessary for creating bicontinuous structures. At polymer concentrations higher and lower than the critical point phase separation is initiated by nucleation and growth and followed by liquid-liquid demixing as the solution passes the binodal and spinodal lines respectively [18, 19]. When the polymer concentration is quenched into the spinodal region phase separation occurs via. spinodal decomposition and bicontinuous domains of polymer rich and polymer poor form. Cellular morphologies can occur with large enough polymer concentrations and faster quenches even if the spinodal is passed [20].

The membrane characteristics can be tailored by adjusting the polymer concentration and cooling rate and for most filtration applications a bicontinuous structure is desired. This is

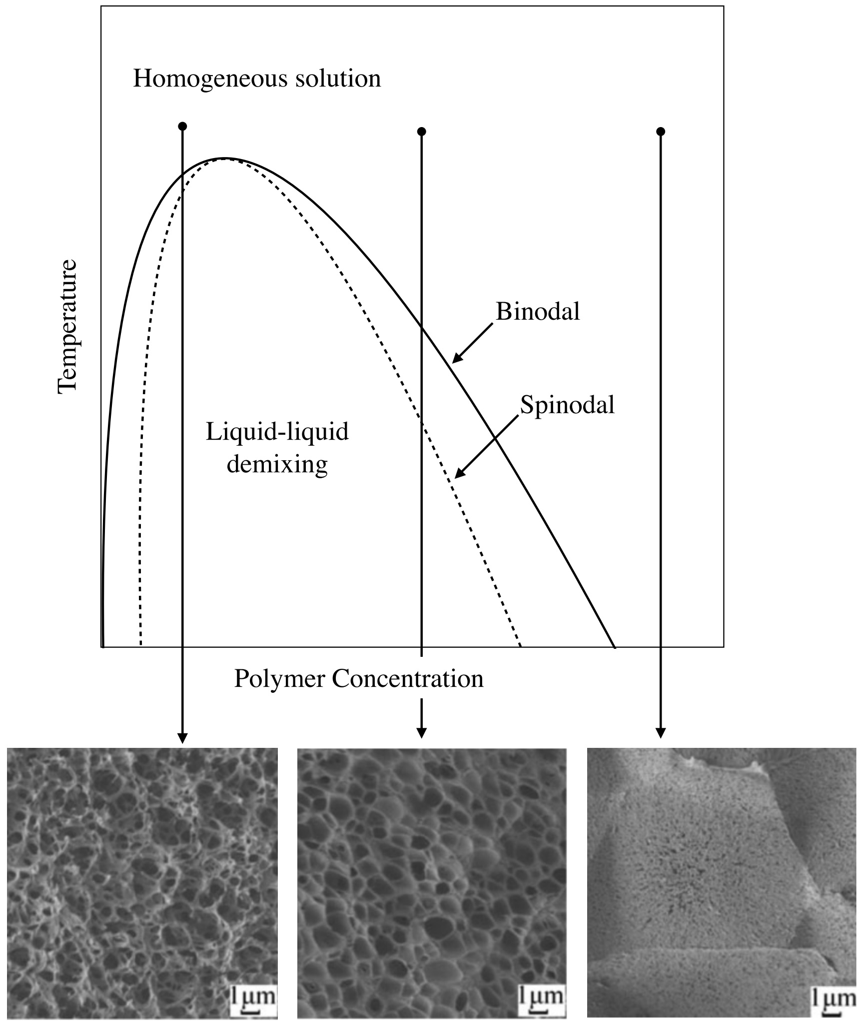

Figure 2.1: Phase diagram for TIPS showing different concentration dependent morphologies. The SEM images are adapted from Lin et al. [20]. These morphologies including (left) bicontinuous, (center) cellular, and (right) spherulitic.

typically achieved using moderate polymer concentrations that allow spinodal decomposition during cooling resulting in a highly porous and bicontinuous structure. As the polymer concentration is increased the pore size decreases and cellular morphologies can begin to form decreasing the interconnectivity of the pores. Pore size can also be decreased by increasing the solvent viscosity and increasing the cooling rate [21–23].

2.1.2 Non-solvent Induced Phase Separation (NIPS)

The NIPS process is typically a ternary system composed of polymer/solvent/non-solvent and may also include additives in either the polymer solution, the coagulation bath, or both.

During NIPS a casted polymer solution is immersed into a non-solvent coagulation bath. This results in solvent leaving the polymer solution and being subsequently replaced with non-solvent and thereby lowers the effectiveness of the solvent resulting in the precipitation of polymer from solution and the formation of a pore network [6]. It is necessary that the solvent and non-solvent be miscible to facilitate their mutual diffusion [24, 25].

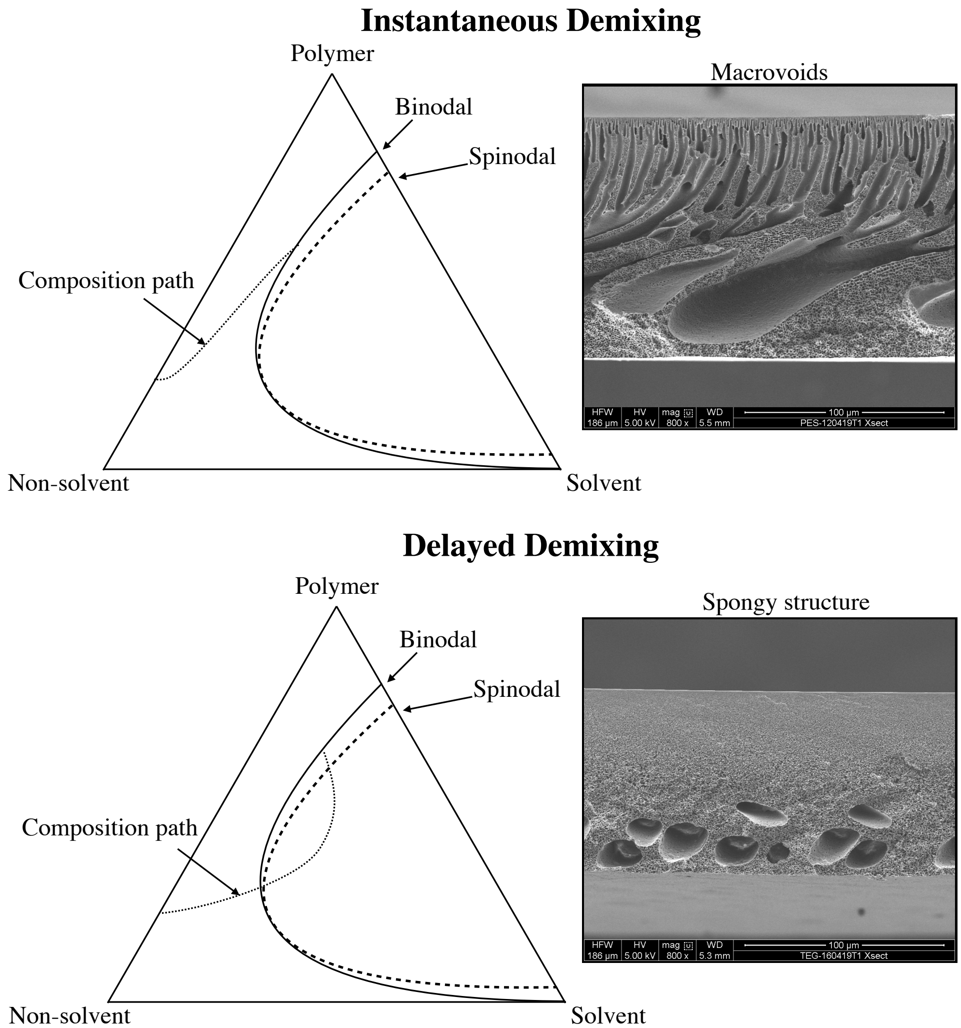

Figure 2.2: Phase diagrams for NIPS showing different composition paths and resulting morphologies for (top) instantaneous demixing and (bottom) delayed demixing. Instantaneous demixing results in large macrovoids (top right) while delayed demixing results in a bi-continuous spongy morphology (bottom right). The membrane forming system for the SEM image on the top right is PES/NMP/Water while the membrane forming system for the bottom right is PES/NMP/PEG/Water and contained a polymeric additive - PEG (polyethylene glycol).

Ternary phase diagrams are used to describe the thermodynamics of phase separation that occurs during NIPS. These phase diagrams are triangular with a single component at each vertex - polymer, solvent, and nonsolvent. The points inside the diagrams represent ternary mixtures of each component. The phase diagrams combined with composition paths that take place during phase inversion can be used to describe instantaneous demixing and delayed demixing: both of which form different membrane structures [26]. These different demixing phenomenon and their resulting membrane structures are represented in figure 2.2. Membranes with large macrovoids and dense top surfaces are typical of instantaneous demixing while asymmetric porous membranes with a spongy morphology are typical of delayed demixing [27, 28]. The formation of macrovoids that form during instantaneous has been extensively studied and they are generally considered unfavorable [28–33]. Additives are used in either the polymer solution, coagulation bath, or both in order to suppress their formation [28, 33–35]. Creating a polymer solution that is initially close to the binodal will typically suppress the formation of macrovoids and result in a spongy porous morphology.

2.2 Modeling the Membrane Formation Process

The length and time scales involved during phase inversion range from molecular to mesoscopic scales and from nanoseconds to several seconds. On the molecular scale short-range interactions between polymer chains, solvents, additives, and non-solvent determine the kinetics and thermodynamics of phase inversion. The mesoscale involves the separation of polymer and solvent domains and how the kinetics and thermodynamics change as the polymer solution undergoes phase inversion. It is important to understand what is happening at both scales as they directly impact the final morphology of a casted membrane [36–43]. Due to the large number of variables and interactions that occur during phase inversion modeling is a useful tool to give understanding and insight to the complex processes occurring.

Many techniques have been used for modeling the membrane formation process. From simple graphical representations of composition paths and morphological evolutions to more complex two- and three-dimensional representations of a membrane structure. The techniques used for modeling the membrane formation process can be broken down into three basic categories of time and length scale, i) macro-scale, ii) molecular-scale, and iii) meso-scale.

2.2.1 Macro-Scale Models

Macro-scale models typically predict temperature and concentration profiles across the thickness of a membrane via Fickian diffusion [44, 45]. Thermodynamic phase diagrams have also been useful for predicting final membrane structures. These phase diagrams are created either experimentally or computationally through the Flory-Huggins free energy of mixing [9–14, 46–48]. These methods give an approximation for the expected membrane morphology given certain casting parameters, however, they do not yield a simulated structure or visual representation of a membrane.

2.2.2 Molecular-Scale Models

Molecular-scale models can give detailed information on the behavior of polymer chains over time. The most useful technique in the molecular-scale is dissipative-particle dynamics (DPD) in which polymer chains are represented by a series of beads connected to each-other and thermodynamic and kinetic details are used to determine their movement. Work done by Wang et al. [49] simulated the NIPS process in two-dimensions tracking the movement of non-solvent into the polymer solution and solvent out of the polymer solution and evaluating the formation process. Extensive work done by Tang et al. [38, 50–55] evaluate different casting conditions for TIPS and NIPS in two- and three-dimensions. The recent work done with DPD captures the separation of polymer and solvent during phase inversion, however, due to the hight computational cost of DPD simulations the domain size is limited to 10s of nm and they are only able to simulate pores of this magnitude.

2.2.3 Meso-Scale Models

Meso-scale modeling is of particular importance because its length and time scales are able to capture pore structures in the order of 10s-100s of μm. The most notable method of modeling phase inversion in the meso-scale is the phase-field method utilizing the Cahn-

Hilliard equation. The Cahn-Hilliard equation accurately describes the diffusive nature of phase inversion and tracks the evolution of multiple components in relation to their kinetic and thermodynamic properties. The first application of the Cahn-Hilliard equation to membrane formation was done by Caneba and Soong [7] in which a one-dimensional domain evaluated the TIPS process and observed pore sizes at a distance away from the cooling surface. Since this work much progress has been made with the speed of computers and the size of domains researchers are able to simulate. The next two sections give an overview of the Phase-Field modeling work done with TIPS and NIPS respectively.

2.2.4 Phase-Field Modeling of Thermally Induced Phase Separation (TIPS)

As previously mentioned Caneba and Soong [7] were the first researchers to use the phase-field method with the Cahn-Hilliard equation to model TIPS. Their model evaluated pore size at different depths from the cooling surface and although their model was only one-dimensional it demonstrated the versatility and promise of this approach. Following this two-dimensional simulations were used to evaluate coarsening and growth rates of the polymer rich and polymer poor (solvent) regions during isotropic quenches [56, 57]. Anisotropic quenches were also studied in one- and two-dimension simulations by Lee et al. [58, 59] where they imposed a temperature gradient onto their domain. This temperature gradient led to an anisotropic pore morphology forming along the gradient direction. Three-dimensional simulations have been done by Mino et al. [60] in which the effect of quenching temperature and a gradient in polymer concentration was investigated. The gradient in polymer concentration models the evaporation of solvent prior to phase inversion and led to an anisotropic pore structure. Lowering the quenching temperature led to smaller overall pores and gave insight to early stage morphology and its effect on domain growth rate.

2.2.5 Phase-Field Modeling of Non-solvent Induced Phase Separation (NIPS)

Phase field modeling for NIPS begins with Barton et al. [61] using two-dimensional simulations to investigate thermodynamic and transport properties of phase separation. They found a significant decrease in structure formation dynamics as polymer concentration increased giving insight to vitrification for systems that undergo a glass transition. Work done by Saxena and Caneba [62] verified the use of a ternary Cahn-Hilliard equation to describe spinodal decomposition with comparisons to experiments. The modeling of NIPS was extended into two and three dimensions by Zhou et al. [63] where the polymer solution interacts with coagulation bath and an anisotropic pore structure during phase inversion. Two- and three-dimensional simulations by Tree et al. [64–66] explore a multi-fluid model for NIPS and investigate the cause of the formation of macrovoids. Tree observes the formation of a dense pore region near the contact surface of the polymer solution and the coagulation bath, however, the formation of macrovoids has yet to be realized by simulations. Work done by Hopp-Hirschler and Nieken [67] utilize two-dimensional domains and a moving phase separation front to evaluate the effect of the front velocity and its effect on pore morphology. They were able to create fingerlike voids - similar to macrovoids - bicontinuous and lamellar morphologies, however, a more rigorous model is needed to verify the mechanism behind the formation of macrovoids.

Bibliography

[1] J. Mulder Marcel Mulder. *Basic Principles of Membrane Technology*. Kluwer Academic Publishers, 1992.

[2] H.K. Lonsdale. Basic principles of membrane technology. *Journal of Membrane Science*, 10:81–181, 1982.

[3] B.S. Laila, V. Kochkodan, R. Hashaikeh, and N. Hailal. A review on membrane fabrication: Structure, properties and performance relationship. *Desalination*, 326:77–95, 2013.

[4] Antoine Venault, Yung Chang, Da-Ming Wang, and Denis Bouyer. A review on polymeric membranes and hydrogels prepared by vapor-induced phase separation process. *Polymer Reviews*, 53:568–626, 2013.

[5] Min Liu, Shenghui Liu, Zhenliang Xu, Yongming Wei, and Hu Yang. Formation of microporous polymeric membranes via thermally induced phase separation: A review. *Frontiers of Chemical Science and Engineering*, 10:57–75, 2016.

[6] Gregory R. Guillen, Yinjin Pan, Minghua Li, and Eric M. V. Hoek. Preparation and characterization of membranes formed by nonsolvent induced phase separation: A re-

view. *Industrial & Engineering Chemistry Research*, 50:3798–3817, 2011.

[7] G.T. Caneba and D.S. Soong. Polymer membrane formation through the thermal-inversion process. 2. mathematical modeling of membrane structure formation. *Macromolecules*, 18:2545–2555, 1985.

[8] G.T. Caneba and D.S. Soong. Polymer membrane formation through the thermal-inversion process. 1. experimental study of membrane structure formation. *Macromolecules*, 18(12):2538–2545, 1985.

[9] D.R. Lloyd, K.E. Kinzer, and H. Tseng. Microporous membrane formation via thermally induced phase separation i. solid-liquid phase separation. *J. Membr. Sci.*, 52:258–267, 1990.

[10] D.R. Lloyd, S.S. Kim, and K.E. Kinzer. Microporous membrane formation via thermally induced phase separation ii. liquid-liquid phase separation. *J. Membr. Sci.*, (1-11), 1991.

[11] S.S. Kim and D.R. Lloyd. Microporous membrane formation via thermally induced phase separation iii. effect of thermodynamic interactions on the structure of isotactic polypropylene membranes. *J. Membr. Sci.*, 64:13–29, 1991.

[12] G. Lim, S.S. Kim, Q. Ye, Y.F. Wang, and D.R. Lloyd. Microporous membrane formation via thermally-induced phase separation iv. effect of isotactic polypropylene crystallization kinetics on membrane structure. *J. Membr. Sci.*, 64:31–40, 1991.

[13] S.S. Kim, G. Lim, A.A. Alwattari, and Y.F. Wang D.R. Lloyd. Microporous membrane formation via thermally-induced phase separation v. effect of diluent mobility and crystallization on the structure of isotactic polypropylene membranes. *J. Membr. Sci.*, 64: 41–53, 1991.

[14] A.A. Alwattari and D.R. Lloyd. Microporous membrane formation via thermally-induced phase separation. vi. effect of diluent morphology and relative crystallization kinetics on polypropylene membrane structure. *J. Membr. Sci.*, 64:55–67, 1991.

[15] D.R. Lloyd and G.B. Lim. Microporous membrane formation via thermally induced phase separation vii. effect of dilution, cooling rate, and nucleating agent addition on morphology. *Journal of Membrane Science*, 79(1):27–34, 1993.

[16] Jeong F. Kim, Ji Hoon Kim, Young Moo Lee, and Enrico Drioli. Thermally induced phase separation and electrospinning methods for emerging membrane applications: A review. *AIChE Journal*, 62:461–490, 2016.

[17] Zhe Liu, Zhenyu Cui, Yuwei Zhang, Shuhao Qin, Feng Yan, and Jianxin Li. Fabrication of polysulfone membrane via thermally induced phase separation process. *Materials Letters*, 195:190–193, 2017.

[18] K. Binder. *Spinodal decomposition in polymers*, volume 5 of *Materials Science and*

Technology, A Comprehensive Treatment. VCH, New York, 1991.

[19] J.W. Cahn. Spindoal decomposition. *Trans. Metall. Soc. AIME*, 242:166, 1968.

[20] Y. Lin, Y. Tang, H. Ma, J. Yang, Y. Tian, W. Ma, and X. Wang. Formation of a bicontinuous structure membrane of polyvinylidene fluoride in diphenyl carbonate diluent via thermally induced phase separation. *J. Appl. Polym. Sci.*, pages 1523–1528, 2009.

[21] J. Yang, D.W. Li, Y.K. Lin, X.L. Wang, F. Tian, and Z. Wang. Formation of a bicontinuous structure membrane of polyvinylidene fluoride in diphenyl ketone diluent via thermally induced phase separation. *Journal of Applied Polymer Science*, 110(1): 341–347, 2008.

[22] X. Li, H. Liu, C. Xiao, S. Ma, and X. Zhao. Effect of take-up speed on polyvinylidene fluoride hollow fiber membrane in a thermally induced phase separation process. *Journal of Applied Polymer Science*, 128(2):1054–1060, 2013.

[23] G-L Ji, C-H Du, B-K Zhu, and Y-Y Xu. Preparation of porous pvdf membrane via thermally induced phase separation with diluent mixture of dbp and dehp. *Journal of Applied Polymer Science*, 105(3):1496–1502, 2007.

[24] M. Guillotin an dC. Lemoyne, C. Noel, and L. Monnerie. Physicochemical processes occuring during formation o fcellulose diacetate membranes - research of criteria for optimizing membrane performance - 4. cellulose diacetate-acetone-organic additive casting solutions. *Desalination*, 21:165, 1977.

[25] D.M. Koenhen, M.H.V. Moulder, and C.A. Smoulders. Phase separation phenomena during formation o fasymmetric membranes. *Journal of Applied Polymer Science*, 21: 199, 1977.

[26] H. Strathmann and K. Kock. Formation mechanism of phase inversion membranes. *Desalination*, 21:141, 1977.

[27] M. Mulder. *Basic Principles of Membrane Technology.* Kluwer Academic Publishers, Dordrecht, The Netherlands, 2003.

[28] C.A. Smolders, A.J. Reuvers, R.M. Boom, and I.M. Wienk. Microstructures in phase-inversion membranes. 1. formation of macrovoids. *Journal of Membrane Science*, 73: 259, 1992.

[29] R. Matz. Structure of cellulose-acetate membranes 1. development of porous structures in anisotropic membranes. *Desalination*, 10:1, 1972.

[30] M.A. Frommer and D. Lancet. *Reverse Osmosis Membrane Research*, chapter The mechanism of membrane formation: Membrane structures and their relation to preparation conditions. Plenum Press: New York, 1972.

[31] H. Strathmann, K. Kock, P. Amar, and R.W. Baker. Formation mechanism of asymmetric membranes. *Desalination*, 16:175, 1975.

[32] R.J. Ray, W.B. Krantz, and R.L. Sani. Linear-stability theory model for finger formation in asymmetric membranes. *Journal of Membrane Science*, 23:155, 1985.

[33] R.M. Boom, I.M. Wienk, T. Vandenboomgaard, and C.A. Smolders. Microstructures in phase inversion membranes 2. the role of a polymeric additive. *Journal of Membrane Science*, 73:277, 1992.

[34] K. Kimmerle and H. Strathmann. Analysis of the structure-tetermining process of phase inversion membranes. *Desalination*, 79:283, 1990.

[35] J.Y. Lai, F.C. Lin, C.C. Wang, and D.M. Wang. Effect of nonsolvent additives on the porosity and morphology of asymmetric tpx membranes. *Journal of Membrane Science*, 118:49, 1996.

[36] Gregory R. Guillen, Guy Z. Ramon, H. Pirouz Kavehpour, Richard B. Kaner, and Eric M.V. Hoek. Direct microscopic observation o fmembrane formation by nonsolvent induced phase separation. *Journal of Membrane Science*, 431:212–220, 2013.

[37] X. He, C. Chen, Z. Jiang, and Y. Su. Computer simulation of formation of polymeric ultrafiltration membrane via immersion precipitation. *J. Memb. Sci.*, 371:108–116, 2011.

[38] Y.D. He, Y.H. Tang, and X.L. Wang. Dissipative particle dynamics simulation on the membrane formation of polymer-diluent system via thermally induced phase separation. *J. Memb. Sci.*, 368:78–85, 2011.

[39] Myeong-Jin Han and Suk-Tae Nam. thermodynamic and rheological variation in polysulfone solution by pvp and its effect in the preparation of phase inversion membrane. *Journal of Membrane Science*, 202:55–61, 2002.

[40] G.D.J. Phillies. Universal scaling equation fro self-diffusion by macromolecules in solution. *Macromolecules*, 19:2367–2376, 1986.

[41] G.D.J. Phillies. The hydrodynamic scaling model for polymer self-diffusion. *J. Phys. Chem*, 93:5029–5039, 1989.

[42] Lin Luo, Tai-Shung Chung, Martin Weber, and Claudia Staudt adn Christian Maletzko. Molecular interactions between acidic sppsu and basic hpei polymers and its effects on membrane formation for ultrafiltration. *Journal of Membrane Science*, 524:33–42, 2017.

[43] Panu Sukitpaneenit and Tai-Shung Chung. Chapter 16 - molecular elucidation of morphology and mechanical properties of pvdf hollow fiber membranes from aspects of phase inversion, crystallization, and rheology. In Tai-Shung Chung and Yingnan Feng, editors, *Hollow Fiber Membranes*, pages 333–360. Elsevier, 2021.

[44] B.F Barton and A.J McHugh. Modeling the dynamics of membrane structure formation in quenched polymer solutions. *Journal of Membrane Science*, 166:119 – 125, 2000.

[45] Dongmei Li, William B. Krantz, Alan R. Greenberg, and Robert L. Sani. Membrane formation via thermally induced phase separation (tips): Model development and validation. *Journal of Membrane Science*, 279:50 – 60, 2006.

[46] C. Qian, S.J. Mumby, and B.E. Eichinger. Phase diagrams of binary polymer solutions and blends. *Macromolecules*, 24:1655–1661, 1991.

[47] P. Flory. *Principles of Polymer Chemistry*. Cornell University Press, 1971.

[48] W.R. Burghardt. Phase diagrams for binary polymer systems exhibiting both crystallization and limited liquid-liquid miscibility. *Macromolecules*, 22:2482–2486, 1989.

[49] X.L. Wang, H.J. Qian, L.J. Chen, Z.Y. Lu, and A. Lin. Dissipative particle dynamics simulation on the polymer membrane formation by immersion precipitation. *J. Memb. Sci.*, 311:251–258, 2008.

[50] Y.H. Tang, Y.D. He, and X.L. Wang. Effect of adding a second diluent on the membrane formation of polymer/diluent system via thermally induced phase separation: Dissipative particle dynamics simulation and its experimental verification. *J. Memb. Sci.*, 409-410:164–172, 2012.

[51] Y.H. Tang, Y.D. He, and X.L. Wang. Three-dimensional analysis of membrane formation via thermally induced phase separation by dissipative particle dynamics simulation. *J. Membr. Sci.*, 437:40–48, 2013.

[52] Y.H. Tang, Y.D. He, and X.L. Wang. Investigation on the membrane formation process of polymer-diluent system via thermally induced phase separation accompanied with mass transfer across the interface. dissipative particle dynamics simulation and its experimental verification. *J. Memb. Sci.*, 474:196–206, 2015.

[53] Y.H. Tang, H.H. Lin, T.Y. Liu, H. Matsuyama, and X.L. Wang. Multiscale simulation on the membrane formation process via thermally induced phase separation accompanied with heat transfer. *J. Memb. Sci*, 515:258–267, 2016.

[54] Han han Lin, Yuan-Hui Tang, Hideto Matsuyama, and Xiao-Lin Wang. Dissipative particle dynamics simulation on the membrane formation of polymer-solvent system via nonsolvent induced phase separation. *Journal of Membrane Science*, 548:288–297, 2018.

[55] Yuan-Hui Tang, Eric Ledieu, M. Rosario Cervellere, Paul C. Millett, David M. Ford, and Xianghong Qian. Formation of polyethersulfone membranes via nonsolvent induced phase separation process from dissipative partilce dynamics simulations. *Journal of Membrane Science*, 599:117826, 2020.

[56] P.K. Chan and A.D. Rey. Computational analysis of spinodal decomposition dynamics

in polymer solutions. *Macromol. Theory Simul.*, 4:873–899, 1995.

[57] B. Barton, P.Graham, and A. McHugh. Dynamics of spinodal decomposition in polymer solutions near a glass transition. *Macromolecules*, 31:1672–1679, 1998.

[58] K. W D. Lee, P.K. Chan, and X. Feng. A computational study into thermally induced phase separation in polymer solutions under a temperature gradient. *Macromol. Theory Simul.*, 11:996–1005, 2002.

[59] K.W.D. Lee, P.K. Chan, and X. Feng. Morphology development and characterization of the phase-separated structures resulting from the thermal-induced phase separation phenomenon in polymer solutions under a temperature gradient. *Chem. Eng. Sci.*, 59: 1491–1504, 2004.

[60] Y. Mino, T. Ishigami, Y. Kagawa, and H. Matsuyama. Three-dimensional phase-field simulations of membrane porous structure formation by thermally induced phase separation in polymer solutions. *J. Membr. Sci*, 483:104–111, 2015.

[61] B.F. Barton and A.J. McHugh. Kinetics of thermally induced phase separation in ternary polymer solutions. i. modeling of phase separation dynamics. *Journal of Polymer Science Part B: Polymer Physics*, 37:1449–1460, 1999.

[62] R. Saxena and G.T. Caneba. Studies of spinodal decomposition in a ternary polymer-solvent-nonsolvent system. *Polymer Engineering and Science*, 42:1019–1031, 2002.

[63] B. Zhou and A.C. Powell. Phase field simulations of early stage structure formation during imersion precipitation of polymeric membranes in 2d and 3d. *J. Membr. Sci.*, 268:150–164, 2006.

[64] D.R. Tree, K.T. Delaney, H.D. Ceniceros, T. Iwama, and G.H. Fredrickson. A multi-fluid model for microstructure formation in polymer membranes. *Soft Matter*, 13:3013–3030, 2017.

[65] D.R. Tree, T. Iwama, K.T. Delaney, J. Lee, and G.H. Fredrickson. Marangoni flows during nonsolvent induced phase separation. *ACS Macro Lett.*, 7:582–586, 2018.

[66] DR Tree, LF Dos Santos, CB Wilson, TR Scott, JU Garcia, and GH Fredrickson. Mass-transfer driven spinodal decomposition in a ternary polymer solution. *Soft Matter*, 15: 4614–4628, 2019.

[67] M. Hopp-Hirschler and U. Nieken. Modeling of pore formation in phase inversion processes: model and numerical results. *J. Memb. Sci.*, 564:820–831, 2018.

Chapter 3

Computational and Experimental Methods

The published papers in chapters 4 and 5 provide sufficient details for reproducible results, however, they did not offer fully detailed explanations of the computational methods. This chapter will take a deeper look into the governing equations of simulating TIPS and NIPS via phase field method using the Cahn-Hilliard equation. It cover the finite difference methods used to calcualte the Cahn-Hilliard equation and updating the simulation going forward in time. A brief discussion on the handcasting of membranes for comparison to experiments presented in Chapter 5 is also discussed.

3.1 Phase-Field Method with the Cahn-Hilliard Equation

The Phase Field method is a useful tool for modeling the membrane formation processes due to its predictive capability for the morphological and structural evolution that occurs during phase separation. The Phase Field method is able to correlate well with time and length scales on which the phase separation occurs [1–6]. The Phase Field method evolves continuous field variables which represent volume-averaged molecular concentrations. These field variables provide details such as the local composition for a particular phase as well as interfaces between unique phases domains where the concentration transitions from one phase to another. This uniquely positions the phase field method when modeling the membrane formation process as large pore networks are able to be realized. The Cahn-Hilliard equation models the diffusive behavior of a conservative field variable that evolves over time making it suitable for modeling membrane formation.

The Cahn-Hilliard equation is a diffusion equation for multi-component mixtures that are governed by a thermodynamic model and free energy of mixing [7]:

$$\frac{\partial \phi_p}{\partial t} = \nabla \cdot \left(M_p \nabla \left(\frac{\partial F_{mix}}{\partial \phi_p} - 2\kappa \nabla^2 \phi_p \right) \right) + \xi, \tag{3.1}$$

The field variable for the polymer volume fraction is ϕ_p and the assumption is made that the leftover concentration is solvent, i.e. $\phi_s = 1 - \phi_p$. Here M_p represents the polymer mobility, F_{mix} is the free energy of mixing, κ is an interfacial energy term and ξ is thermal fluctuations. The thermal fluctuations are added at each time step to ensure an accurate behavior of the simulations and unique solutions for each time step.

3.1.1 Free energy of Mixing

The free energy of mixing used for both TIPS in chapter 4 and NIPS in chapter 5 is a binary Flory-Huggins free energy of mixing [8]:

$$F_{mix} = k_b T \left[\frac{\phi_p}{N} \ln \phi_p + \phi_s \ln \phi_s + \chi \phi_p \phi_s \right] \tag{3.2}$$

where k_b is the Boltzmann constant, T is temperature in Kelvin, ϕ_p is the polymer volume fraction, N is the number of repeating units in a polymer, ϕ_s is the solvent volume fraction and in the simulations the substitution for solvent $\phi_s = 1 - \phi_p$ is made, and χ is the interaction parameter between polymer and solvent. The interaction parameter χ can be dependent on many factors including temperature and composition. For modeling TIPS the interaction parameter χ is temperature dependent and a more thorough discussion is outlined in Section 3.1.3. The interaction parameter used for modeling NIPS was dependent on the local water concentration and a more thorough discussion is outlined in Section 3.1.4.

3.1.2 Polymer Mobility

The polymer mobility is determined from the free energy of mixing and polymer diffusivity by the equation [9]:

$$M_p = \frac{D_p}{\partial^2 F_{mix} / \partial \phi_p^2} \tag{3.3}$$

The diffusion of polymer in solution is described by the Phillies model [10, 11]:

$$D_p = D_0 \exp(-\alpha c^{\nu}) \tag{3.4}$$

Where D_0 is the diffusivity of polymer in solvent in the dilute limite and α and ν are scaling coefficients. The diffusivity of a polymer chain is temperature sensitive which is accounted for by the Einstein relationship:

$$D_0 = \frac{k_b T}{f}, \tag{3.5}$$

where k_b is the Boltzmann constant, T is temperature in Kelvin, and f is a friction coefficient. With these three equations it is possible to have a temperature and concentration sensitive mobility which more closely resembles the physical relationship observed with polymer solutions.

3.1.3 Modeling Thermally Induced Phase Separation (TIPS)

This section outlines the main equations used to model TIPS in chapter 4. Isotropic quenching was used to characterize the model and observe the transition from bicotinuous/spinodal decomposition to cellular/nucleation and growth. The system of interest PVDF/DPC is an upper critical solution temperature system so the domain is initialized at a high temperature above the critical temperature and is cooled to equilibrium. In the case of isotropic quenching the entire domain was cooled at a linear rate in the following way:

$$T(t) = T_i - (T_i - T_f) \times \left(\frac{t}{\Omega}\right) \tag{3.6}$$

Here T_i is the initial temperature, T_f is the final temperature, t is the current simulation time and Ω is the total simulation time.

Anisotropic quenching is a more realistic representation of the physical cooling process that occurs during TIPS. This was achieved my using the solution to a 1D thermal diffusivity equation. The polymer solution is initially homogeneous and at an elevated temperature before being brought into contact with a cooling surface. The following equation was used to model this thermal diffusion:

$$T(t, x) = T_{bath} + (T_i - T_{bath}) \times \operatorname{erf}\left(\frac{x}{2\sqrt{\alpha_T t}}\right) \tag{3.7}$$

where T_{bath} is the temperature of the cooling surface, T_i is the initial domain temperature. x is distance away from the cooling surface, α_T is the thermal diffusivity, and t is the time. In this way a thermal gradient can be modeled and propagated through the domain.

The interaction parameter χ in equation 3.2 was tied to temperature via the equation [12, 13]:

$$\chi = \frac{425}{T} - 0.338 \tag{3.8}$$

where T is the solution temperature in Kelvin. This temperature dependent interaction parameter leads to phase separation as the domain is cooled into the unstable regions within the phase diagram. The phase diagram calculated with the FH free energy of mixing in Eq. 3.2 with the above interaction parameter in Eq. 3.8 has a critical temperature around 460 K and the system of interest PVDF/DPC has a crystallization temperature of approximately 390 K [12]. Therefore the simulations are initialized at a temperature of 460 K and then cooled. For isotropic quenches the domain is cooled to a temperature of 390 K and the anisotropic quenches are cooled to 273 K, 298 K, and 333 K respectively. To model the rapid drop in diffusivity that occurs during phase change below the crystallization temperature polymer diffusivity is reduced by a factor of 10^3.

3.1.4 Modeling Non-solvent Induced Phase Separation (NIPS)

This section outlines the main equations used to model NIPS in chapter 5. The specific system of interest is Polyethersulfone/N-methyl-2-pyrrolidone/Water (PES/NMP/H_2O). A constant coagulation bath composition was imposed on the top surface of the domain subsequently diffuses into the domain via Fick's second law:

$$\frac{\partial f_N}{\partial t} = \nabla \cdot \left(D_N(\phi_P)\nabla f_N\right), \tag{3.9}$$

where f_N is the fraction of non-solvent and D_N is the nonsolvent diffusivity which was dependent on polymer concentration ϕ_p. To Phillies equation was again used to quantify non-solvent diffusion in relation to polymer concentration:

$$D_N = D_N^o exp(-\alpha c^\nu) \tag{3.10}$$

whwere D_N^o is the diffusivity of water in NMP. The interaction parameter in NIPS is related to the stability of polymer in the solvent and to show this relationship a weighted linear average for the interaction parameter was used:

$$\chi = f_N \chi_{PN} + (1 - f_N)\, \chi_{PS}, \tag{3.11}$$

The interaction parameters used were for the PES/NMP/Water system at 25 °C and $\chi_{PN} = 1.5$ is the interaction parameter for PES and non-solvent (water) and $\chi_{PS} = 0.034$ is the interaction parameter for PES and NMP [14–16]. In this way as the non-solvent diffuses into the polymer solution NMP becomes dilluted and the solution becomes unstable resulting in PES precipitation out of solution.

3.2 Numerical Solution Techniques

The methodology for simulating both SIPS and NIPS follows a sequence of calculations. The calculation and update of the CH equation is similar for both methods, however, the NIPS method involves updating f_N. In this section we'll add the subscript $_ijk$ to values that are unique to each grid point in the domain $1 \le i \le nx$, $1 \le j \le ny$, and $1 \le k \le nz$. The series of calculations for computing the CH equation working from the innermost parenthesis $\left(\frac{\partial F_{mix,ijk}}{\partial \phi_{p,ijk}} - 2\kappa \nabla^2 \phi_{p,ijk}\right)$ and working outward until the entire equation is complete:

1. Apply boundary conditions and compute the laplacian of polymer concentration $\nabla^2 \phi_{p,ijk}$, multiply the laplacian by 2κ, then calculate the $\frac{\partial F_{mix,jik}}{\partial \phi_{p,ijk}}$. Subtract $2\kappa\nabla^2\phi_{p,ijk}$ from the first derivative of free energy.

2. Store the the value from step 1 for each grid point, $\mu_{ijk} = \left(\frac{\partial F_{mix,ijk}}{\partial \phi_{p,ijk}} - 2\kappa \nabla^2 \phi_{p,ijk}\right)$

3. Compute the mobility $M_{p,ijk}$ for each grid point and store the value

4. Compute the non-uniform laplacian for $M_{p,ijk}$ and μ_{ijk}

5. Perform Euler update and add random fluctuations in concentration

To compute the laplacian in equation in step 1 a finite difference scheme is used as follows:

$$\nabla^2 \phi_{p,ijk} \approx \frac{\phi_{i-1jk} + \phi_{i+1jk} + \phi_{ij-1k} + \phi_{ij+1k} + \phi_{ijk-1} + \phi_{ijk+1} - 6.0\phi_{ijk}}{h^2} \tag{3.12}$$

where h is the grid size when $\Delta x = \Delta y = \Delta z$. When taking the divergence of equation 3.1 with a non-uniform mobility term the finite difference approximation is a bit more tedious. In this way we first calculate the laplacian for the ijk directions separately then add them together:

$$\nabla \cdot M_i \nabla \mu_i \approx \frac{\mu_{i-1jk} b_{i-1jk} + \mu_{i+1jk} b_{i+1jk} - (b_{i-1jk} + b_{i+1jk})\mu_{ijk}}{h^2} \tag{3.13}$$

$$\nabla \cdot M_j \nabla \mu_j \approx \frac{\mu_{ij-1k} b_{ij-1k} + \mu_{ij+1k} b_{ij+1k} - (b_{ij+1k} + b_{ij+1k})\mu_{ijk}}{h^2} \tag{3.14}$$

$$\nabla \cdot M_k \nabla \mu_k \approx \frac{\mu_{ijk-1} b_{ijk-1} + \mu_{ijk+1} b_{ijk+1} - (b_{ijk-1} + b_{ijk+1})\mu_{ijk}}{h^2} \tag{3.15}$$

$$\nabla \cdot M_{ijk} \nabla \mu_{ijk} \approx (\nabla \cdot M_i \nabla \mu_i) + (\nabla \cdot M_j \nabla \mu_j) + (\nabla \cdot M_k \nabla \mu_k) \tag{3.16}$$

And here the variable b is calculated for the 6 nearest neighbors of our grid point at ijk. Here is the calculation for $i-1, j, k$:

$$b_{i-1jk} = \frac{2.0}{1.0/M_{i-1jk} + 1.0/M_{ijk}} \tag{3.17}$$

and this is repeated five more times at locations $i+1jk$, $ij-1k$, $ij+1k$, $ijk-1$, and $ijk+1$ to account for the 6 nearest neighbors positions.

For NIPS we also need to update the local concentration of non-solvent $f_{N,ijk}$ by calculating the non-uniform Laplacian and updating the concentrations as we march forward in time as in steps 3-5 above. To update the non-solvent field we first apply boundary conditions and compute the non-uniform laplacian of $\nabla D_{N,ijk} \cdot \nabla f_N$ via equation 3.13, where μ_{ijk} is simply the local concentration of non-solvent $f_{N,ijk}$ and $D_{N,jik}$ is substituted in for M_{ijk}.

To advance the simulations in time for evolving the polymer and nonsolvent domains equation 3.1 and 3.9 is updated via an Euler update. For updating the the polymer domain the following update is done:

$$\phi_p^{t+1} = \phi_p^t + \Delta t \left[\nabla \cdot \left(M_p \nabla \left(\frac{\partial F_{mix}}{\partial \phi_p} - 2\kappa \nabla^2 \phi_p \right) \right) + \xi \right] \tag{3.18}$$

The non-solvent concentration is also evolved using an Euler update:

$$f_N^{t+1} = f_N^t + \Delta t \left(\nabla D_N \cdot \nabla f_N \right) \tag{3.19}$$

In both methods for SIPS and NIPS periodic and no-flux boundary conditions are imposed. In the case of isotropic quenches in section 3.1.3 periodic boundary conditions are used on all sides of the simulation domain. Both the NIPS model and the anisotropic thermal quenches for TIPS use no-flux boundary conditions on the top and bottom of the domain with PBC in the other directions. The no-flux boundary conditions are imposed at the boundaries of the domain and this is achieved by setting the out of boundary positions equal to the current position when calculating the laplacian in equations 3.12 and 3.13.

3.3 Simulation Length and Time Scales

The simulation and time length scale are directly related to the diffusivity used in the determination of the mobility term M in equation 3.1 and consequently the polymer diffusivity. The resolution between polymer rich and polymer poor regions take approximately 4-7 grid spaces in both methods. The grid size needs to resolve the interface between polymer and solvent while still maintaining a mean field approximation. For the simulations a time constant was assumed and from there a grid size constant was calculated from the following diffusion relationship:

$$D = \frac{cm^2}{s} = \frac{h^2}{\tau} \tag{3.20}$$

Where τ is the time constant calculated by dividing the total number of simulation time steps taken by the time step size. For example a simulation length of $2,000,000$ timesteps and a step size $dt = 0.005$ you have a total simulation length of $10,000\tau$. Next you take the value for polymer diffusivity $D = 1.0 \times 10^{-7}\frac{cm^2}{s}$ and taking a $h = 35nm$ by using the relationship from equation 3.20 we can back calculate the step size:

$$\tau = \frac{(35nm)^2}{1.0 \times 10^{-7}\frac{cm^2}{s}} = 0.1225ms$$

And for this simulation the total time is $nstep = 2,000,000 = 10,000\tau = 1.225s$.

3.4 Simulation Analysis

The simulation pore size in Chapters 4 and 5 were analyzed using algorithms that measured the distance between polymer rich regions and polymer poor regions. The algorithm did one-dimensional scans in the $y-z$ plane in both the y and z direction and then calculated the average pore size of the standard deviation between each scan. This builds a layer-by-layer 2D slice of pore size which can then be plotted to quantify the pore size in relation to the depth of a membrane. This analysis was used for the anisotropic TIPS simulations and the three-dimensional NIPS simulations and the code is located in the Appendix for reference. The algorithm also calculates the average pore size for the entire domain by taking an average of each layer-by-layer pore size and then calculating the standard deviation of pore size for the entire domain. This 3D pore size was used to analyze isotropic thermal quenches employed in the TIPS method. A simpler algorithm was created to analyze the two dimensional simulations in Chapter 5. It calculates pore size similar to the previous algorithm, however, due to it's two-dimensional nature only one scan per layer was done. This algorithm is presented in the Appendix.

The Hoshen-Kopelmann algorithm is used to determine the interconnectivity of the pores [17]. Pores labeled numerically as the algorithm encounters them. If a new point is a nearest neighbor to another known pore then the new point assumes the same label. After a complete scan of the domain is made the pore labels are re-arranged by size, the largest pores having a label 1. The continuity parameter Γ_c is then calculated by dividing the size of the largest pore by the total size of all pores. When a domain is completely bi-continuous Γ_c will be 1 since the largest pore is the only pore present. When the domain exhibits a more cellular morphology the continuity parameter begins to decrease. Domains with a swiss cheese like structure have very low values of Γ_c meaning that the size of the largest pore is only a small fraction of the total amount of pores. The algorithm for calculating the continuity of the domain and creating a labeled visualization of the pores is in the Appendix.

3.5 Handcasting Membrane Samples

Handcast membranes were made during an internship at MilliporeSigma to compare with the NIPS model in Chapter 5. These membranes were composed of PES/NMP/PVP/Water system. The polymer solution was made from 15 wt% BASF PES E3010, 10 wt% PVP k90 and NMP. The addition of PVP was used to increase solution viscosity and to promote bi-continuous morphology [18–20]. The coagulation bath consist of different vol% NMP and DI water and the three concentrations of NMP were added to the coagulation bath - 0, 20, and 40 vol% NMP. The coagulation bath was heated to a temperature of 50 °C and the casting surface was also heated to 50 °C. The polymer solution was cast onto a plastic film over a glass plate and then moved to the coagulation bath. The glass plate was left in the coagulation bath for 30 seconds then the plastic film with membrane attached was cut from the glass plate and moved to a bath of 15 °C DI water for 24 hours. After soaking for 24 hours the membranes were then placed dried inside of aluminium frames at 70 °C to reduce shrinking. The membranes were then imaged via SEM to evaluate the top surface and cross section morphology.

Bibliography

[1] L.Q. Chen. Phase-field models for microstructure evolution. *Annual review of materials research*, 32:113–140, 2002.

[2] W.J. Boettinger, J.A. Warren, C. Beckermann, and A Karma. Phase-field simulation

of solidification. *Annual review of materials research*, 32:163–194, 2002.

[3] I. Steinbach. Phase-field models in materials science. *Model. Simul. Mat. Sci. Eng.*, 17: 1–31, 2009.

[4] S. Asai, S. Majumdar, A. Gupta, K. Kargupta, and S. Ganguly. Dynamics and pattern formation in thermally induced phase separation of polymer-solvent system. *Comput. Mater. Sci.*, 47(193-205), 2009.

[5] L.T. Yan and X.M. Xie. Numerical simulation of substrate effects on spinodal decomposition in polymer binary mixture: Effects of the surface potential. *Polymer*, 47: 6472–6480, 2006.

[6] K. W D. Lee, P.K. Chan, and X. Feng. A computational study into thermally induced phase separation in polymer solutions under a temperature gradient. *Macromol. Theory Simul.*, 11:996–1005, 2002.

[7] J.W. Cahn and J.E. Hilliard. Free energy of a nonuniform system. i. interfacial free energy. *J. Chem. Phys.*, 20:256–267, 1958.

[8] P. Flory. *Principles of Polymer Chemistry.* Cornell University Press, 1971.

[9] B. Barton, P.Graham, and A. McHugh. Dynamics of spinodal decomposition in polymer solutions near a glass transition. *Macromolecules*, 31:1672–1679, 1998.

[10] G.D.J. Phillies. Universal scaling equation fro self-diffusion by macromolecules in solution. *Macromolecules*, 19:2367–2376, 1986.

[11] G.D.J. Phillies. The hydrodynamic scaling model for polymer self-diffusion. *J. Phys. Chem*, 93:5029–5039, 1989.

[12] Y. Lin, Y. Tang, H. Ma, J. Yang, Y. Tian, W. Ma, and X. Wang. Formation of a bicontinuous structure membrane of polyvinylidene fluoride in diphenyl carbonate diluent via thermally induced phase separation. *J. Appl. Polym. Sci.*, pages 1523–1528, 2009.

[13] Y.H. Tang, H.H. Lin, T.Y. Liu, H. Matsuyama, and X.L. Wang. Multiscale simulation on the membrane formation process via thermally induced phase separation accompanied with heat transfer. *J. Memb. Sci*, 515:258–267, 2016.

[14] Gabriel Tkacik and Leos Zeman. Component mobility analysis in the membrane forming system water/n-methyl-2-pyrrolidone/polyethersulfone. *Journal of Membrane Science*, 31:273–288, 1987.

[15] Leos Zeman and Gabriel Tkacik. Thermodynamic analysis of a membrane-forming system water/n-methyl-2-pyrrolodone/polyethersulfone. *Journal of Membrane Science*, 36:119–140, 1988.

[16] Li Xu and Feng Qui. Simultaneous determination of three floryehuggins interaction parameters in polymer/solvent/nonsolvent systems by viscosity and cloud point measurements. *Polymer*, 55:6795–6802, 2014.

[17] S. Frijters, T. Kruger, and J. Harting. Parallelised hoshen-kopelman algorithm for lattice-boltzmann simulations. *Comput. Phys. Commun.*, 189:92, 2015.

[18] Jeong Rim Hwang, Seong-Hoe Koo, Jong-Ho Kim, Akon Higuchi, and Tae-Moon Tak. Effects of casting solution composition on performance of poly(ether sulfone) membrane. *J. Appl. Polym. Sci.*, 60:1343–1348, 1996.

[19] Seong Hyun Yoo, Jong Hak Kim, Jae Young Jho, Jongok Won, and Yong Soo Kang. Influence of the addition of pvp on the morphology of asymmetric polyimide phase inversion membranes: efect of pvp molecular weight. *Journal of Membrane Science*, 236:203–207, 2004.

[20] M. Amirilargani, E. Saljoughi, T. Mohammadi, and M.R. Moghbeli. Effects of coagulation bath temperature and polvinylpyrrolidone content on flat sheet asymmetric polyethersulfone membranes. *Polymer Engineering and Science*, pages 885–893, 2010.

Chapter 4

Paper 1: Mesoscopic simulations of thermally-induced phase separation in PVDF/DPC solutions

4.1 Abstract

We present a phase-field model of thermally-induced phase separation in polymer solutions, calibrated for the polyvinylidene fluoride (PVDF)/diphenyl carbonate (DPC) system. Large-scale three-dimensional computer simulations were performed for isotropic and anisotropic thermal quenches, and the evolution and structure of the resulting two-phase morphology is analyzed. Isotropic quenches, in which the temperature is uniformly reduced below the binodal temperature, were conducted to understand the initiation and coarsening of the polymer-rich and polymer-poor phases throughout time. Anisotropic quenches, in which the system is cooled from one particular surface, were also conducted to understand how gradients in the characteristic domain size develop for varying conditions. In these anisotropic quenches, we observe the formation of a dense skin layer adjacent to the cooling surface, the thickness of which depends on several parameters including the polymer volume fraction, the assumed bath temperature that is maintained at the cooling surface, and the rate of thermal conduction through the polymer solution. The model here can be adapted to other polymer/solvent systems by modifying the thermodynamic and kinetic parameters specific to the two species.

4.2 Introduction

Porous polymer membranes are typically fabricated from a polymer solution that is made to undergo an internal phase separation process [1–4]. This process is thermodynamically driven by either a change in temperature (known as thermally-induced phase separation, TIPS) or the introduction of a second solvent (known as solvent-induced phase separation, SIPS) that creates a miscibility gap leading to the co-formation of a polymer-rich phase and a polymer-poor phase. Subsequently, the polymer-poor phase is removed to form a dispersion of internal porosity, and the polymer-rich phase is solidified to form a membrane. Various geometries can be achieved with TIPS and/or SIPS processing including flat sheets and hollow fibers. The critical characteristic of the system is the morphology of the internal pores, which can vary significantly depending on the composition of the solution as well as the processing conditions. Predicting the size of the pores, the uniformity of the pore size, and the continuity/discontinuity of the pore structure is challenging, and requires consideration of both the thermodynamic and kinetic interactions between the constituent species within the solution throughout the phase separation process.

Considering the large number of variables, the complexity of the phase separation process, and the cost of parametric experimental studies, computer simulations play an important role in the on-going goal of engineering customizable membrane structures with specific pore size distributions. Various simulation techniques have been proposed and applied in the literature, falling into three broad categories based on their representative length scale: (i) molecular-scale simulations, (ii) meso-scale simulations, and (iii) macro-scale simulations.

Macro-scale simulations generally utilize continuum transport models (i.e., Fickian diffusion of mass and heat) to predict temperature and concentration profiles along entire membrane cross-sections [5, 6]. Predictions of pore sizes can then be made using thermodynamic models such as Flory-Huggins theory as well as kinetic theories to describe the phase coarsening rates at particular temperatures. This approach does not explicitly track the dynamics of the polymer and solvent species, nor does it track the formation of the two-phase morphology. However, it does provide approximations of porosity variations on a membrane-wide scale.

On the other end of the spectrum, molecular-scale simulations such as molecular dynamics (MD) can predict detailed information of individual polymer chains, such as their conformational changes through time. Such details however are only possible for system sizes on the order of 10's of nanometers (or, perhaps 1 - 20 polymer chains), as MD simulations track every atomic trajectory. Coarse-grained models such as dissipative particle dynamics

(DPD) reduce this level of detail while still capturing chain dynamics with a point-mass representation. The work of Wang et al. [7], He et al. [8], and Tang et al. [9–12] demonstrate the ability of DPD simulations to capture the formation and evolution of polymer-rich and polymer-poor phase domains during TIPS and/or SIPS processing. Due to computational demands, DPD simulations are limited to system sizes of approximately 10 nm, and therefore can only capture pore size distributions on this scale.

Simulation methods at intermediate length scales are particularly promising as they resolve features at the level of the pore network. These techniques capture the evolution of the phase separation process with variables stored on a computational grid that define the two distinct phases. Termonia [13–15] developed a Monte Carlo (MC) lattice diffusion model to simulate the coagulation process during SIPS, revealing various morphologies including fingerlike pores depending on the coagulation rate. He et al. [16] employed a similar technique and analyzed the pore structure as a function of polymer content. These stochastic MC studies were performed on two-dimensional lattices.

Deterministic meso-scale simulations have also been developed, most notably phase-field models utilizing the Cahn-Hilliard (CH) equation. The CH equation is essentially a diffusion equation for multi-component mixtures that is informed by an assumed thermodynamic model for the energy of mixing, which may induce phase separation. Caneba and Soong [17] demonstrated the earliest application of the CH equation to specifically simulate the polymer membrane formation process. They conducted one-dimensional (1D) simulations of the TIPS process in a polymer-solvent system at various locations relative to a cooling surface, using the Flory-Huggins and free-volume theory models for the thermodynamic and kinetic descriptions, respectively. Their results estimated pore sizes as a function of membrane depth away from the cooling surface, thus demonstrating the versatility of this approach. However, although 1D simulations can provide predictions of pore size, they do not offer information regarding the continuity/discontinuity of a porous network.

Soon after, two-dimensional (2D) simulations of TIPS in polymer-solvent systems were reported [18, 19] that assumed isotropic quenches focusing in particular on the growth and coarsening rates of the polymer-rich and polymer-poor phases. The effect of temperature gradients (i.e. anisotropic quenches) were studied by Lee et al. [20, 21] and Kukadiya et al. [22] with 1D and then 2D simulations. Recently, Mino et al. [23] conducted three-dimensional simulations of the TIPS process, including the effects of a polymer concentration gradient that leads to an anisotropic structure. The SIPS process has also been simulated with phase-field models [24–27] to investigate the coagulation process. Due to the significant hydrodynamic transport processes associated with SIPS during the exchange of the two solvents, recents efforts to simulation SIPS have employed fluid-based simulation methods, including the lattice-Boltzmann method [28] and the multi-fluid model of Tree et al. [29, 30].

In this work, we have utilized a CH model to simulate the TIPS process in three dimensions for both isotropic and anisotropic quenches. We have chosen the PVDF/DPC polymer/solvent system which is commonly used in TIPS membrane processing [31, 32]. The size and interconnectivity of the pore structures are analyzed for varying polymer volume fractions and temperature quench rates. In our anisotropic quench simulations, we observe the formation of a dense skin layer, as observed in experiments [33], whose thickness is found to depend on the bath temperature and the thermal conductivity of the polymer solution.

4.3 Methods

The CH equation employed here evolves in space and time a conserved field variable, ϕ_p, representing the local polymer volume fraction in a solution. We assume a binary solution of polymer and solvent, hence $\phi_p + \phi_s = 1$ at any location, and only ϕ_p is required to represent the system. The equation is given by:

$$\frac{\partial \phi_p}{\partial t} = \nabla \cdot \left(M_p \nabla \left(\frac{\partial F_{mix}}{\partial \phi_p} - 2\kappa \nabla^2 \phi_p \right) \right) + \xi, \tag{4.1}$$

where M_p is the temperature- and concentration-dependent polymer mobility, F_{mix} is the free energy of mixing between polymer and solvent, κ is a term that scales the interfacial energy between the polymer-rich and polymer-poor phases, and ξ is a random number centered at zero associated with thermal fluctuations. The CH equation is essentially a diffusion equation which we are applying to investigate a liquid-liquid phase separation process. We acknowledge that the model does not account for convective mass transport. However, unlike the SIPS process that involves long-range fluid transport of solvent species which justifies a fluid model [28–30], the TIPS process is a more local redistribution of polymer and solvent during quenching in a very viscous system, thus diffusion is the dominant mode of transport.

The Flory-Huggins (FH) free energy of mixing of a polymer-solvent system is used for F_{mix}:

$$F_{mix} = k_b T \left[\frac{\phi_p}{N} \ln \phi_p + \phi_s \ln \phi_s + \chi \phi_p \phi_s \right] \tag{4.2}$$

where the substitution ϕ_s = 1 - ϕ_p is made, N is the degree of polymerization set to a value of $N = 150$, and χ is the polymer-solvent interaction term, which is temperature dependent and expressed by:

$$\chi = \frac{425}{T} - 0.338, \tag{4.3}$$

where T is assumed to be in Kelvin. The values used in this equation have been shown previously to be appropriate for the PVDF/DPC system [12, 31]. The binary phase diagram of PVDF/DPC is depicted in Fig. 4.1, and the FH energy curves are shown in the sub-plot. We have also included images of small 2D simulations with our model, demonstrating the variations in morphology with T and ϕ_p. As ϕ_p is increased, the morphology transitions from discrete droplets of the polymer-rich phase, to a bicontinuous morphology, to discrete droplets of the polymer-poor phase. In the PVDF/DPC system, the critical temperature calculated using Flory-Huggins with the above interaction parameter and degree of polymerization is approximately 460 K and the crystallization temperature is approximately 390 K [31]. Hence, we only present the phase diagram in this temperature range, which is associated with the

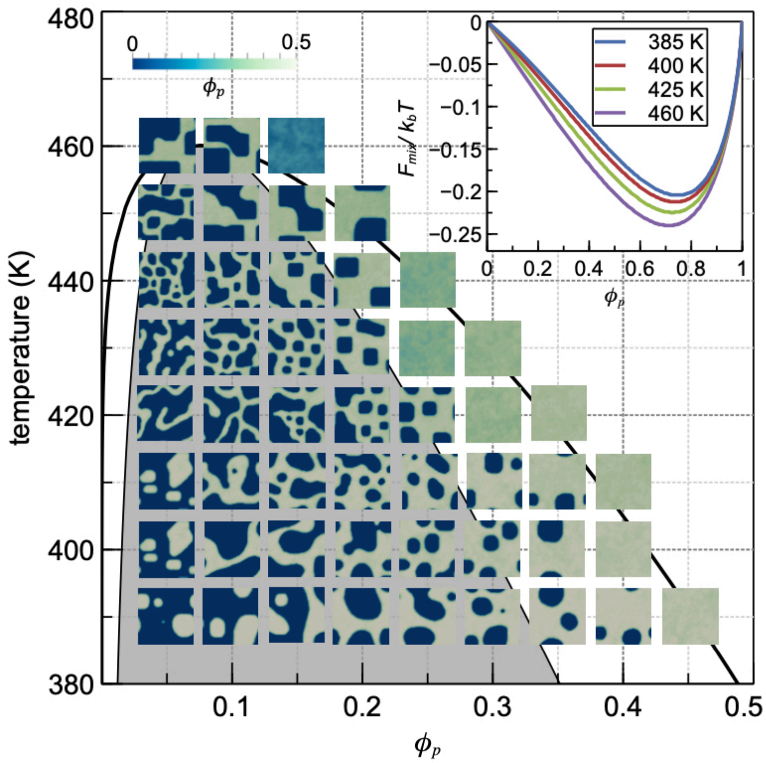

Figure 4.1: Phase diagram for PVDF/DPC superimposed with simulation images of isothermal quenches at different temperatures and ϕ_p. Various morphologies appear including droplets of the polymer-rich phase, bicontinuous domains of both phases, and droplets of the polymer-poor phase. The inset plot shows the Flory-Huggins free energy of mixing, Eq. (4.2), for four different temperatures.

liquid-liquid phase separation that occurs during TIPS. In our simulations, we do not observe phase separation in regions above the binodal line on the phase diagram, as expected. In Eq. 4.1, we assign $\kappa = 0.5$ which is chosen to keep the diffuse interface widths at 5-7 grid spacings (ideal for the CH model), and ξ is a random number chosen in the interval [-0.1,0.1].

The polymer mobility M_p is closely related to the self-diffusivity of a polymer chain in solution D_p according to the relationship [19]:

$$M_p = \frac{D_p}{\partial^2 F_{mix}/\partial \phi_p^2}. \tag{4.4}$$

The polymer self-diffusivity is highly dependent on temperature and the local ϕ_p. An experimentally-validated model developed by Phillies [34, 35] is used here to describe the dependence of D_p on ϕ_p:

$$D_p = D_0 \exp(-\alpha c^{\nu}) \tag{4.5}$$

where D_o is the diffusivity of a single chain in an infinitely dilute solution, and c is the polymer concentration in g/L calculated by $c = \phi_p \frac{M_v}{M_w}$ where M_v is the molar volume and M_w is the molar weight of the monomer in the polymer chain. We use values of $M_v = 38.2$ mL/mol and $M_w = 64.03$ g/mol, suitable for a monomer of PVDF [36]. The parameters α and ν are system-dependent scaling coefficients which are generally fit according to experimental data. Diffusion data specific to PVDF/DPC was not found in the literature, so we choose values of $\alpha = 0.2$ and $\nu = 0.4$, which fall in reasonable bounds for many other polymer solutions [34, 35]. The diffusivity of a single polymer chain in a dilute solution is given by the Einstein equation:

$$D_0 = \frac{k_b T}{f}, \tag{4.6}$$

where f is a friction factor. Equations (4.5) and (4.6) together account for the temperature- and concentration-dependence of D_p.

The self-diffusivity of PVDF in DPC has not been reported, hence we assume a value of 1×10^{-7} cm^2/s at the critical temperature of 460 K. Figure 4.2 plots the diffusivity versus ϕ_p at three temperatures: $T = 460$ K, 391 K and 389 K. The inset of Fig. 4.2 plots the polymer mobility M_p versus ϕ_p. At $T = 460$ K, the decrease in polymer diffusivity with increasing ϕ_p is accounted for by Eq. (4.5). Diffusivities at temperatures below 460 K can be obtained by linearly scaling the diffusivity at 460 K, according to Eq. (4.6). However, at the crystallization temperature of 390 K, the diffusivity can be expected to abruptly drop due to the state change. To account for this, at temperatures below 390 K, we reduce the diffusivity by a factor of 10^3 (i.e., we divide the calculated D_p by 1000). Hence, in Fig. 4.2 we plot diffusivities at 391 K (just above the crystallization temperature) and 389 K (just

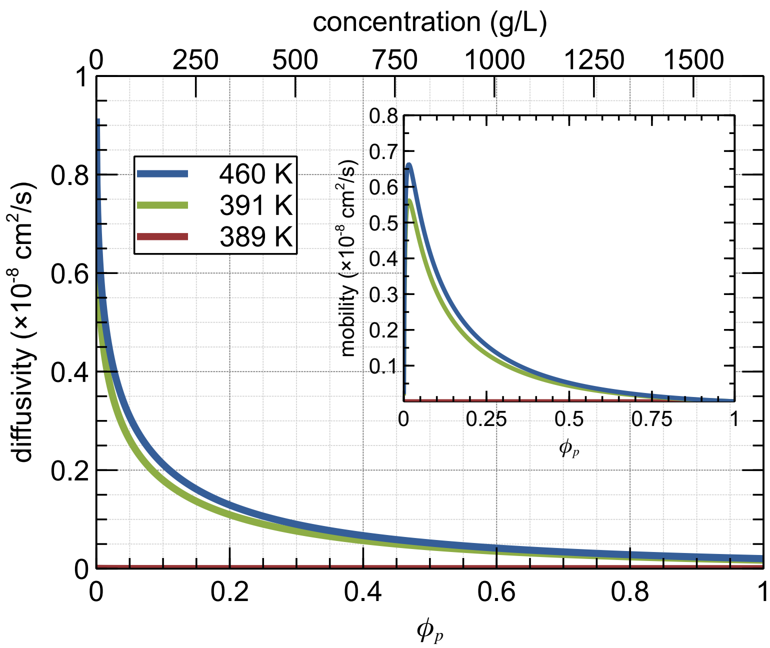

Figure 4.2: Polymer diffusivity versus ϕ_p calculated by Eq. (4.5) for three temperatures - two above the crystallization temperature and one below. The inset plot shows the polymer mobility versus ϕ_p for the same three temperature.

below the crystallization temperature), showing the abrupt drop in values.

To execute our simulations, we solve Eq. (4.1) with a straightforward explicit finite difference scheme on a rectilinear grid with uniform spacing between nodes. To reduce round-off error, the equations are solved with reduced units of length ($\bar{l}$) and time ($\bar{t}$), whereby the grid spacing $\Delta x = 1\ \bar{l}$ and the time step size is $\Delta t = 0.005\ \bar{t}$, a value that ensures numerical stability. The reduced diffusivity is set equal to unity at $T = 460$ K, and linearly scaled for temperatures below that, taking into account the reduction below the crystallization temperature. In all simulations herein, the maximum temperature is 460 K, assigned as the initial temperature, followed by either an isotropic or anisotropic quench in which the temperature is reduced through time. Following the simulations, we convert all length and time scales back into physical units by assuming $\bar{l} = 35$ nm (hence, each grid node represents a box with side lengths of 35 nm) and $\bar{t} = \bar{l}^2/10^{-7}\text{cm}^2/\text{s} = 1.225\ \times 10^{-4}$ s. Our choice of $\bar{l}$ is somewhat arbitrary, however in order to satisfy the mean-field representation of the polymer solution, it should be larger than the chain radius of gyration, hence the physical length and time herein are relevant for such a condition. Choosing a different $\bar{l}$ will effectively re-scale the physical time duration for the simulations.

Isotropic quenching was conducted by reducing the temperature uniformly at a constant linear rate throughout the entire domain:

$$T(t) = T_i - (T_i - T_f) \times \left(\frac{t}{\Omega}\right) \tag{4.7}$$

where T_i is the initial temperature (always set to 460 K), T_f is the final temperature, t is the current simulation time, and Ω is the total simulation time. Different quench rates were achieved by varying Ω. In our isotropic quenches, T_{final} was set to the crystallization temperature, and periodic boundary conditions were applied in all three directions.

Anisotropic quenching was also performed to more accurately capture the effects of temperature gradients on the phase separation process, ultimately leading to anisotropic pore morphologies. For anisotropic quenching, we assumed a uniform, initial temperature of 460 K. One surface of the domain (at $x = 0$) was kept at a constant cool temperature, and the temperature profile is obtained from the solution to a 1D, semi-infinite heat equation:

$$T(t, x) = T_{bath} + (T_i - T_{bath}) \times \operatorname{erf}\left(\frac{x}{2\sqrt{\alpha_T t}}\right) \tag{4.8}$$

where T_{bath} is the temperature of the cool surface, assumed to be in contact with a bath, and α_T is the thermal diffusivity of the polymer solution (in units of cm^2/s), assumed to be uniform and equal in both the polymer-rich and polymer-poor phases. To relate the thermal diffusivity and the polymer diffusivity, and make the results more general, we utilize the non-dimensional Lewis number defined as $Le = \frac{\alpha_T}{D_0}$ [37]. We used several values of Le to determine its effect on pore morphology, including $Le = 50$, 100, and 150. In our anisotropic quenches, periodic boundary conditions were applied in the y- and z-directions, while no-flux boundaries (for ϕ_p) were applied in the x-direction. We also used three different bath temperatures, $T_{bath} = 273$ K, 298 K, and 333 K, and analyzed its effect on pore morphology.

We implemented a variety of analysis tools to evaluate the nature of the pore morphology. First, the average pore size was calculated by conducting one-dimensional sweeps along columns of grid points in each direction and calculating the average distance between interfaces within the polymer-poor phase (with an interface being defined as a location where $\phi_p = 0.25$). The average distance between interfaces was then averaged for all columns of grid points in the x-, y-, and $z-$ directions. Second, we evaluated the interconnectivity of the porosity using a Hoshen-Kopelman (HK) cluster counting algorithm [38]. The HK algorithm identifies and labels individual domains of a phase, in our case the polymer-poor phase associated with the porosity. We then compute a continuity parameter Γ_c that represents the continuity of the pore network in space [39]. This parameter is calculated by dividing the volume of the largest pore (V_L) by the total volume of all the porosity (V_T):

$$\Gamma_c = \frac{V_L}{V_T}. \tag{4.9}$$

The value of Γ_c quantifies the interconnectivity of the pore network, whereby $\Gamma_c = 1$ represents the case where all the porosity is associated with a single pore (complete interconnectivity). Otherwise, as Γ_c approaches zero, the largest pore is a small fraction of the total porosity, which is thus discrete and discontinuous.

All simulations were executed on 16-core CPU nodes, parallelized by domain decomposition along the x-direction. Each simulation typically required a few hours of wall time to complete.

4.4 Results

4.4.1 Isotropic Quench

First, we performed isotropic quenches in which the temperature throughout the simulation domain was decreased uniformly from an initial value of 460 K to a final value of 390 K (i.e., the crystallization temperature). We conducted three-dimensional simulations with grid sizes of 256 $\times$ 256 $\times$ 256, corresponding to 9 $\times$ 9 $\times$ 9 μm. Figure 4.3 depicts the evolution of the two-phase morphology through time. The system is initialized by assigning

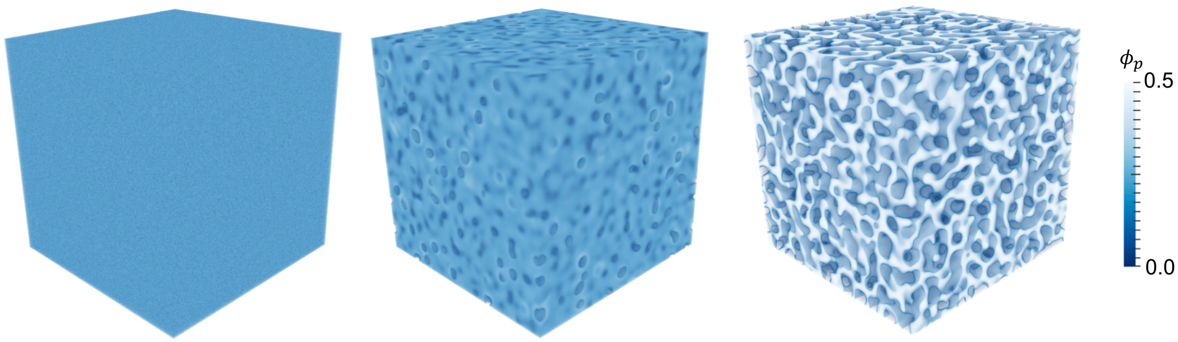

Figure 4.3: Progressive snapshots in time of an isotropic quench simulation with an initial temperature of 460 K and a final temperature of 390 K. Here, the average volume fraction is $\bar{\phi}_p = 0.15$, the quench rate is 56 K/s, and the domain size is $9 \times 9 \times 9$ μm. The white regions correspond with the polymer-rich phase, and the blue semi-transparent regions correspond with the polymer-poor phase. The final structure on the right depicts a bicontinuous morphology.

values of ϕ_p about an average polymer volume fraction $\bar{\phi}_p$ with an initial random variability of $\pm$ 0.05. The images in Fig. 4.3 correspond to $\bar{\phi}_p = 0.15$ and a quench rate of 56 K/s. At the initial temperature, the solution is fully soluble and there is no thermodynamic driving force for phase separation. As temperature decreases, a continuous change in F_{mix} leads to phase separation along with a continuous change in M_p that governs the rate of phase separation. Polymer-rich and polymer-poor domains form, and the local polymer content in these domains continuously changes during quenching according to the binodal line of the phase diagram. At the end of the simulation, the domain has a temperature of 390 K, and a two-phase morphology exists in which the polymer-rich domains have ϕ_p values of approximately 0.5 and the polymer-poor domains have ϕ_p values very near zero.

In our isotropic quench study, we varied two key parameters: $\bar{\phi}_p$ and the quench rate. Throughout the simulations, we computed the average pore size defined as the average distance between interfaces through the polymer-poor phase. This data is shown in Fig. 4.4. The left plot shows the average pore size versus temperature (which is analogous with time for our isotropic quenches) with $\bar{\phi}_p = 0.15$ for four different quench rates: 56, 70, 93, and 140 K/s . The right plot shows the average pore size at the end of the quench versus $\bar{\phi}_p$ for the same quench rates. Overall, we see a trend in which a higher quench rate (i.e.,

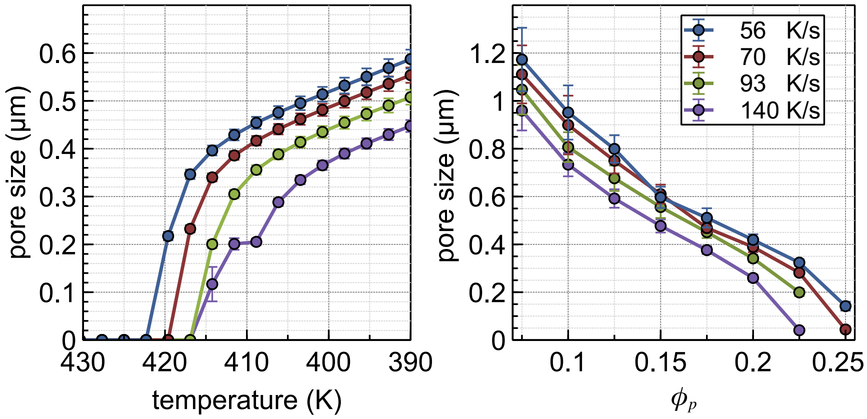

Figure 4.4: Data plots of: (left) average pore size versus temperature during isotropic quenches for $\bar{\phi}_p = 0.15$ and four different quench rates, (right) average pore size versus $\bar{\phi}_p$ at the end of the quenches for four different quench rates. The average pore size decreases with increasing quench rate and polymer volume fraction.

a faster quench) results in a smaller pore size. There are two factors accountable for this relationship. First, lower quench rates allow more time for the two-phase morphology to coarsen. A lower quench rate permits the system to remain at higher temperatures (yet, below the binodal line) where the polymer mobility is higher for longer times. Second, a higher quench rate results in a delay in the onset of phase separation, due to a delay in overcoming the nucleation barrier. We see evidence of this in the left plot of Fig. 4.4, where the data points depart from the x-axis at different temperatures (or, equivalently, different times). Lower quench rates allow more time for nucleation to occur when the system first crosses the binodal line. Due to these two effects, at the end of the quench, the pore size is larger for lower quench rates.

In addition, when examining the right plot of Fig. 4.4, we see that the pore size is highly sensitive to the average polymer fraction. This is to be expected, given that the porosity is derived from the polymer-poor phase, the quantity of which is determined from a tie line using the lever rule with a fulcrum at $\bar{\phi}_p$. There is also a secondary factor associated with the polymer mobility. The average mobility in the system as a whole is higher for smaller values of $\bar{\phi}_p$, due to the fact that D_p decreases exponentially with ϕ_p. Hence, the coarsening rate will be higher for lower values of $\bar{\phi}_p$.

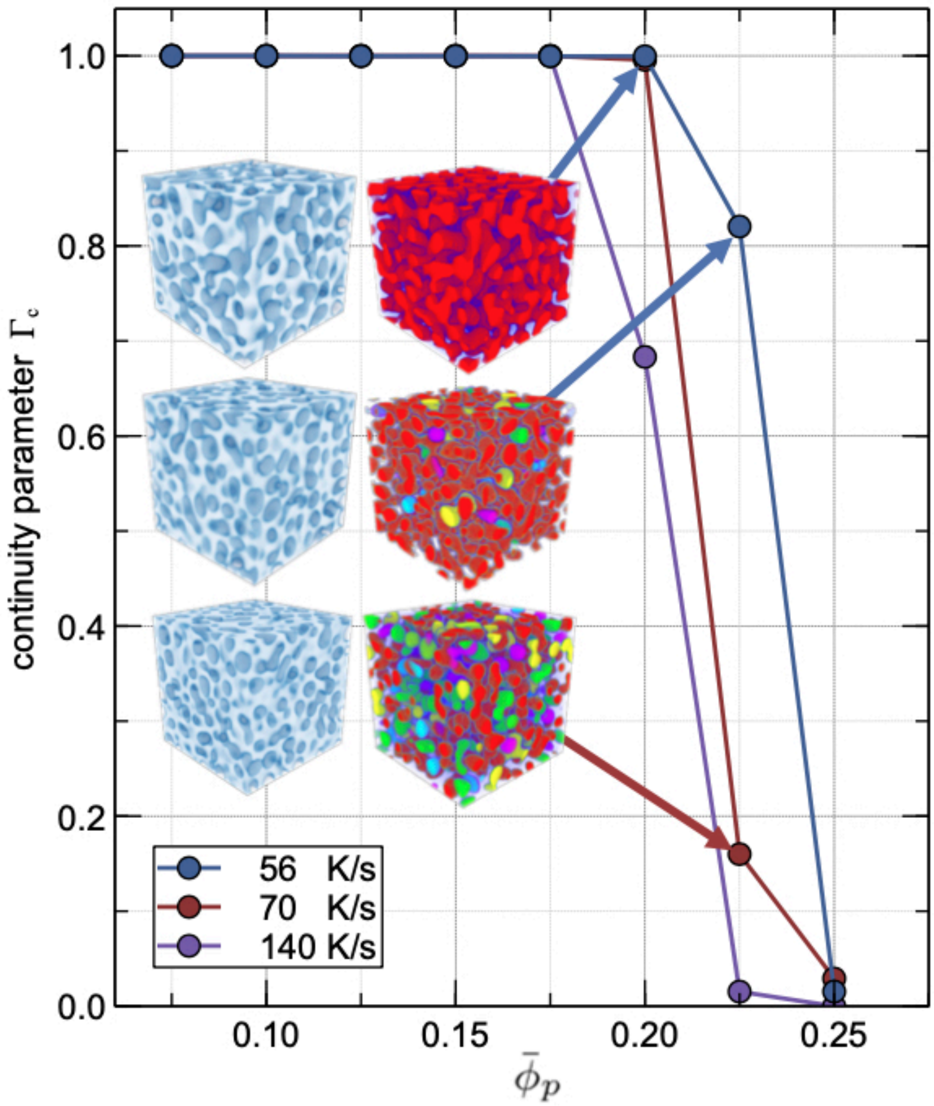

Figure 4.5: The calculated continuity parameter Γ_c versus $\bar{\phi}_p$ for three different quench rates. Increasing $\bar{\phi}_p$ leads to a transition in morphology from a fully continuous pore network (Γ_c = 1) to a highly discontinuous pore network ($\Gamma_c << 1$). The inset images show simulation snapshots of the polymer fraction on the left and the pore regions on the right colored according to pore size (red = large pores and violet = small pores). Individual pore domains are identified by the HK algorithm.

The continuity of the porosity for these systems was also computed using the HK algorithm discussed above. Figure 4.5 plots the continuity parameter Γ_c versus $\bar{\phi}_p$ for three quench rates. For values of $\bar{\phi}_p$ below 0.175, the porosity is completely interconnected, as evident by values of $\Gamma_c = 1$. Within the plot, we added images of different structures at the end of their quenches. The blue-white images on the left depict the polymer volume fraction. The multi-colored images on the right depict the porous regions in the domain (i.e. the polymer-poor phases). The pores are shaded according to their respective volume, with a red shading corresponding to a large pore volume and a blue shading corresponding to a small pore volume. These images show the transition from an interconnected porosity to a discrete disconnected porosity, which abruptly occurs in the range $0.175 < \bar{\phi}_p < 0.25$. The interconnectivity of the pore network is inherently important to the separation performance of polymer membranes. The quench rate was seen to influence the pore morphology only within a range of polymer volume fractions ($\bar{\phi}_p = 0.17$ - 0.25). Below this range the system is within the spinodal region and the relatively equal quantities of polymer-rich and polymer-poor phases strongly favor a bicontinuous structure. Above this range, the polymer-poor phase is a minority phase and forms discontinuous droplets regardless of quench rate.

4.4.2 Anisotropic Quench

The temperature quenching that occurs in an actual TIPS processing procedure occurs in an anisotropic manner. One of the surfaces of the polymer-solution is brought into contact with a cooling bath, which leads to a one-dimensional heat transfer process. Anisotropic quenching can therefore lead to anisotropic pore structures, as the local change in temperature versus time depends strongly on the distance from the cooling surface within the polymer solution.

To investigate an anisotropic quench process, we conducted simulations in which the $x = 0$ surface was held at a constant temperature correlating with the temperature of the bath, T_{bath}. The rest of the domain was assigned the initial temperature, $T_i = 460$ K. Equation (4.8) was then solved within the domain to determine the local temperature at a specific

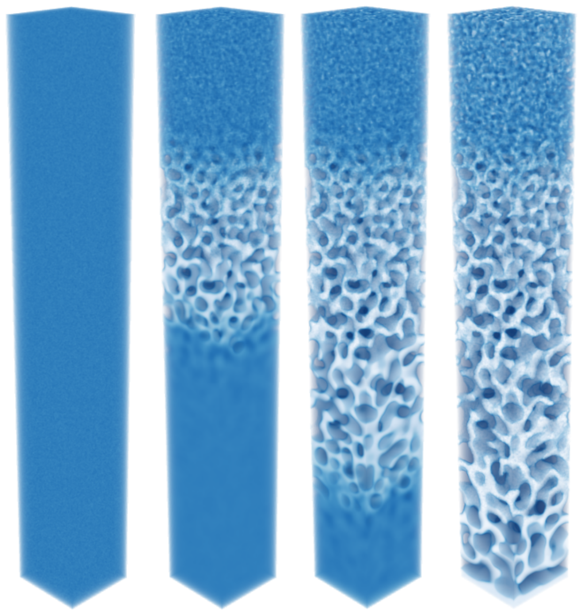

Figure 4.6: Progressive snapshots of an anisotropic quench simulation which is cooled from the top surface that is held at a constant temperature $T_{bath} = 298$ K. The one-dimensional heat flux leads to a non-uniform decrease in temperature according to Eq. (4.8), and a resulting non-uniform pore network. The images were taken at times of 0s, 0.179 s, 0.358 s, and 0.894 s (the final time). The domain size is $17.5 \times 2.45 \times 2.45$ μm, the polymer volume fraction is $\bar{\phi}_p = 0.15$, and the Lewis number is $Le = 50$.

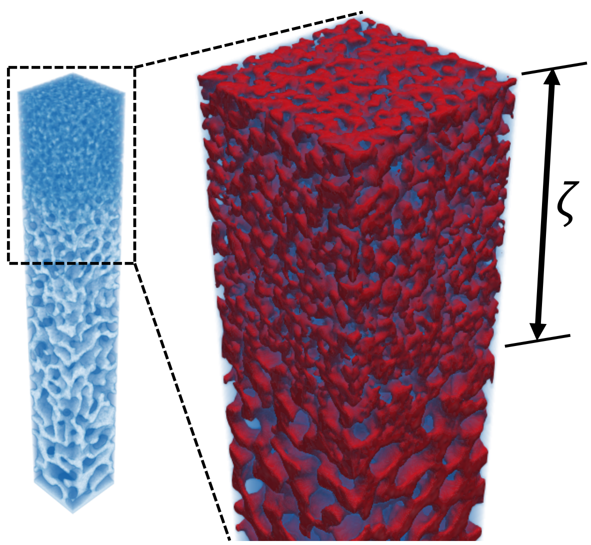

Figure 4.7: A close-up image of the dense skin layer near the cooling surface. An isosurface (drawn at $\phi_p = 0.15$) is included to better visualize the structure. Within the skin layer, a small degree of phase separation has occurred, which was essentially halted early in the simulation due to the local temperature dropping below the crystallization temperature.

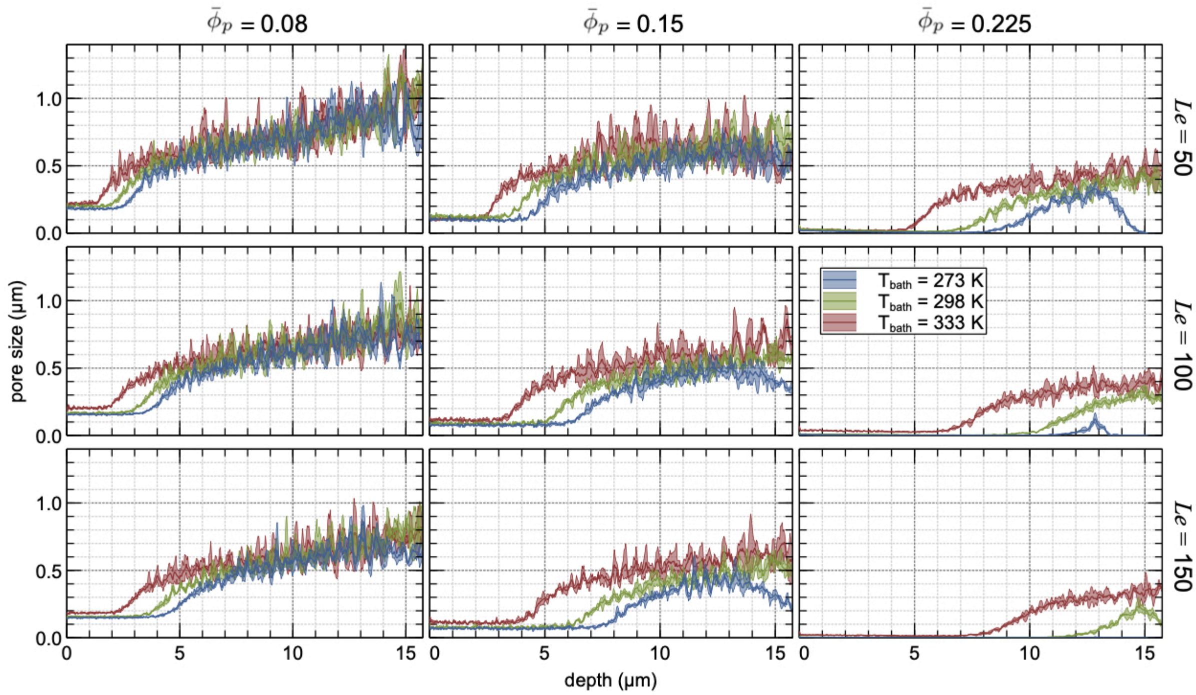

Figure 4.8: The computed values of pore size versus depth below the cooling surface (in the x-direction) for all of the conditions tested in the anisotropic quench study. The left column of plots corresponds to $\bar{\phi}_p = 0.08$, the middle column to $\bar{\phi}_p = 0.15$, and the right column to $\bar{\phi}_p = 0.225$. The top row of plots corresponds to $Le = 50$, the middle row to $Le = 100$, and the bottom row to $Le = 150$. Within each plot, there are three data lines corresponding with three bath temperatures, as indicated.

point in space and time. Our anisotropic quenches differ from those of Mino et al. [23] by the fact that we utilize a non-uniform and time-dependent temperature field, whereas Mino et al. assumed an initial polymer concentration gradient (to represent a preliminary solvent evaporation) followed by an isotropic temperature quench.

We elongated the domain in the x-direction, and the overall grid sizes used in this section were 500 × 70 × 70, corresponding with $17.5 \times 2.45 \times 2.45$ μm. Figure 4.6 shows snapshots of the anisotropic phase-separation process at progressive instances in time for a polymer volume fraction of $\bar{\phi}_p = 0.15$. The initial temperature of the polymer solution was $T_i = 460$ K, the bath temperature was $T_{bath} = 298$ K. To relate the mass diffusivity of the polymer with the thermal diffusivity of heat in the material, we utilize the dimensionless Lewis number defined as $Le = \frac{\alpha_T}{D_o}$ [37] (again, D_o is the diffusivity of a polymer chain in a dilute solution at 460 K). For the images in Fig. 4.6, $Le = 50$. In Fig. 4.6, the final simulation time is 0.894 seconds, which is the required time to reduce the temperature throughout the domain to a value below the crystallization temperature, 390 K. This time span is perhaps shorter than that occurring in laboratory TIPS processing, a result of the fact that the depth of our simulation domain is less than the thickness of a typical polymer membrane (e.g. 170 μm).

The anisotropic quenching clearly results in a gradient in pore size in the x-direction. Most notably, at the top of the domain near the cooling surface, there exists a region where complete phase separation has not fully occurred. Within this region, the temperature dropped quickly below the crystallization temperature, and very little time was available for the phase-separation process. This region, however, does not have a completely uniform polymer concentration. Figure 4.7 shows a close-up view of the structure, with an isosurface drawn to depict $\phi_p = 0.15$, which is the average polymer fraction. Clearly, variations in the polymer fraction exist in this region, corresponding therefore to a distribution of very small pores. Polymer-rich and polymer-poor domains have begun to develop in this region, but have not fully evolved to their preferred values of ϕ_p. This dense layer near the cooling surface represents a skin layer which is commonly observed in polymer membranes [40]. A recent experimental work [41] reveals very similar pore structures resulting from the TIPS process, including a dense skin layer and a gradient in pore size in the direction perpendicular to the skin surface.

To quantify these anisotropic pore structures, we have computed the average pore size versus depth from the cooling surface along the x-direction. This was performed using the same procedure as described above, however only columns of data along the y and z directions were probed, and the average pore size for each x plane was computed. This data is plotted in Fig. 4.8 versus depth away from the cooling surface. Here, we have varied three critical parameters: $\bar{\phi}_p$, Le, and T_{bath}. Figure 4.8 contains a 3 $\times$ 3 array of plots in which each column represents a particular value of $\bar{\phi}_p$, each row represents a particular value of Le, and each line within the plots represents a particular value of T_{bath}. The three bath temperatures

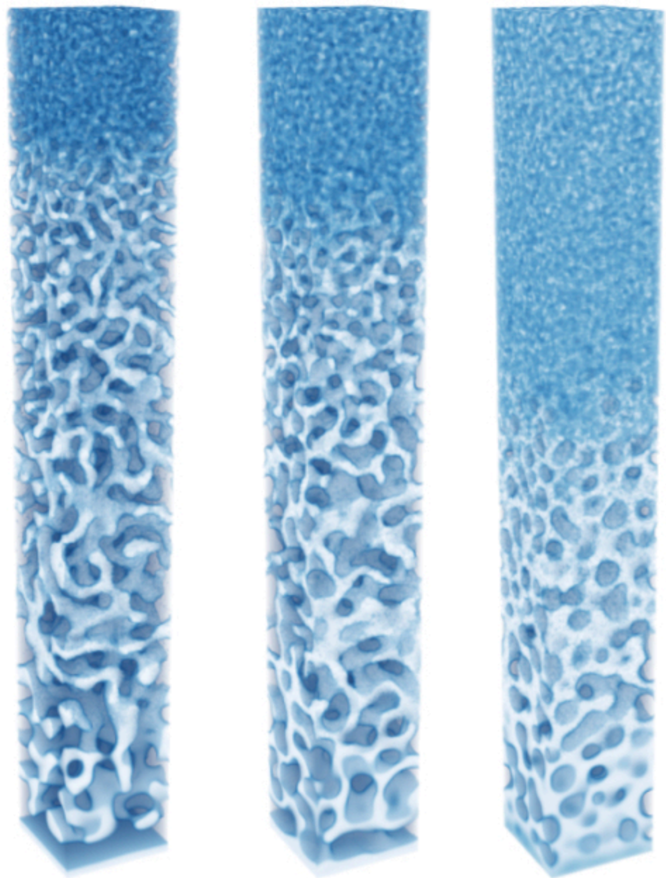

Figure 4.9: Images from anisotropic quench simulations for three different polymer volume fractions: (left) $\bar{\phi}_p = 0.08$, (middle) $\bar{\phi}_p = 0.15$, and (right) $\bar{\phi}_p = 0.225$. These images show the variation of skin layer depth versus $\bar{\phi}_p$, as well as the difference in the interconnectivity of the pore structures with $\bar{\phi}_p$. $T_{bath} = 298$ K and $Le = 50$.

chosen correspond to ice water ($T_{bath} = 273$ K), room temperature ($T_{bath} = 298$ K), and a hotter temperature ($T_{bath} = 333$ K).

The plots in Fig. 4.8 show several important relationships. First, the skin layer can be recognized by the small values of pore size at small depths. The pore size increases rather abruptly at the bottom of the skin layer. Below the skin layer, the pore size increases somewhat gradually along the depth of the system. Overall, an observed trend is that the pore size is larger for smaller values of $\bar{\phi}_p$, smaller values of Le, and higher values of T_{bath}. Higher bath temperatures and smaller values of Le result in lower temperature drop rates, hence longer periods of phase separation and therefore larger pore sizes. Smaller

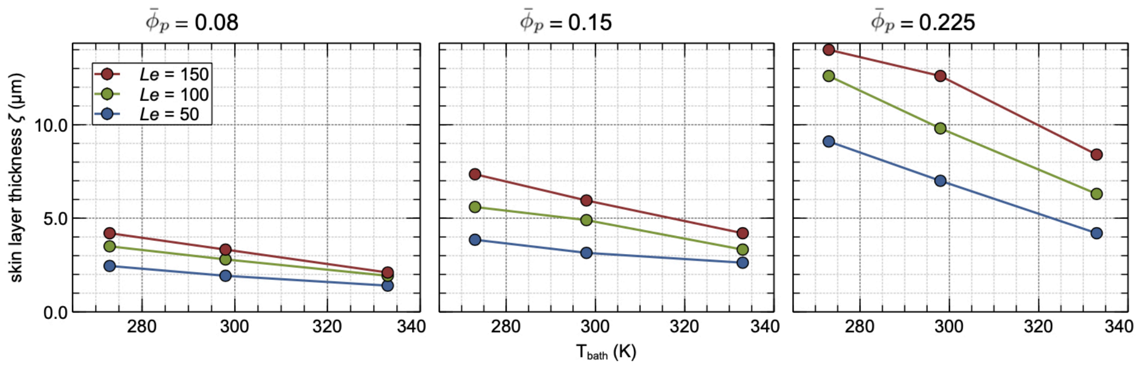

Figure 4.10: The skin layer thickness versus T_{bath} for the three different $\bar{\phi}_p$ values tested and the three different Le numbers tested. The skin layer thickness increases with decreasing T_{bath}, increasing $\bar{\phi}_p$, and increasing Le.

values of $\bar{\phi}_p$ correspond with less polymer content and larger pore sizes, as observed for the isotropic quenches. In Fig. 4.8, cases where the pore size apparently drops at the largest depths (around 15 μm) is actually due to insufficient time to initiate phase separation in those regions. Longer simulations would be required to capture phase separation in those regions. Figure 4.9 shows the final states of the three different $\bar{\phi}_p$ values used, illustrating the difference in the size and interconnectivity of porosity. The $\bar{\phi}_p = 0.08$ sample has the largest pore sizes and a very interconnected pore network. Conversely, the $\bar{\phi}_p = 0.225$ sample has smaller pores which appear to be discrete and non-interconnected.

We measured the skin layer thickness, and plotted these values versus T_{bath} for the three $\bar{\phi}_p$ values and the three Le values, as shown in Fig. 4.10. These values of skin layer thickness were arbitrarily defined as the depths at which the average pore size increased by 50% in Fig. 4.8. We observe that skin thickness increases with decreasing T_{bath}, increasing Le, and increasing $\bar{\phi}_p$. These results are generally in agreement with experiments [33]. The skin layer thickness is also highly dependent on the crystallization temperature, which we kept at a constant value of 390 K, corresponding with the PVDF/DPC system [31]. This anisotropic quench process illustrates the intricate relationships between many thermodynamic, kinetic, and processing conditions that ultimately govern the resultant pore structure that forms.

4.5 Conclusion

To fully capitalize on the power of computer simulations in predicting membrane morphology for a particular material system and processing conditions, three-dimensional simulations and analysis must be performed. With current computer power, and with appropriate mesoscale models, simulations are now able to predict the complex networks of porosity that form within polymer membranes during fabrication. In this paper, we present a phase-field model to investigate the TIPS process through time for the PVDF/DPC material system. We conducted both isotropic and anisotropic quenches, and analyzed the pore networks for both average pore size and pore interconnectivity. In our anisotropic quenches, the domain size in the direction of heat flux was 17.5 μm, which is an order of magnitude smaller than typical polymer sheet membranes. Hence, a current simulation cannot predict pore structure throughout an entire membrane cross-section. Nevertheless, important insights can be made from this simulation model, which we contend is fast becoming a valuable tool for membrane manufacturers.

The current model does not capture convective transport of the polymer species within the solution during phase separation. A diffusion-based model is likely more suited to the TIPS process than the SIPS process, whereby the exchange of two solvent species occurs on longer length scales. A next step will therefore be to investigate the SIPS process, likely with an expanded model that captures flow due to interfacial forces in the system.

4.6 Acknowledgements

The authors would like to acknowledge funding from the MAST Center (Membrane Research Science and Technology Center) through NSF (project no. IPP 1361809). The authors would also like to acknowledge the Arkansas High Performance Computing Center for the computational resources. Thanks as well to mentors from MilliporeSigma and 3M for guidance and insight to the direction of the project.

Bibliography

[1] P. van de Witte, P.J. Dijkstra, J.W.A. van den Berg, and J. Feijen. Phase separation processes in polymer solutions in relation to membrane formation. *Journal of Membrane Science*, 117(1):1 – 31, 1996.

[2] Gregory R. Guillen, Yinjin Pan, Minghua Li, and Eric M. V. Hoek. Preparation and characterization of membranes formed by nonsolvent induced phase separation: A review. *Industrial & Engineering Chemistry Research*, 50:3798–3817, 2011.

[3] Agnieszka K. Holda and Ivo F.J. Vankelecom. Understanding and guiding the phase inversion process for synthesis of solvent resistant nanofiltration membranes. *Journal of Applied Polymer Science*, 132:42130, 2015.

[4] Jeong F. Kim, Ji Hoon Kim, Young Moo Lee, and Enrico Drioli. Thermally induced phase separation and electrospinning methods for emerging membrane applications: A review. *AIChE Journal*, 62:461–490, 2016.

[5] B.F Barton and A.J McHugh. Modeling the dynamics of membrane structure formation in quenched polymer solutions. *Journal of Membrane Science*, 166:119 – 125, 2000.

[6] Dongmei Li, William B. Krantz, Alan R. Greenberg, and Robert L. Sani. Membrane formation via thermally induced phase separation (tips): Model development and validation. *Journal of Membrane Science*, 279:50 – 60, 2006.

[7] X.L. Wang, H.J. Qian, L.J. Chen, Z.Y. Lu, and A. Lin. Dissipative particle dynamics simulation on the polymer membrane formation by immersion precipitation. *J. Memb. Sci.*, 311:251–258, 2008.

[8] Y.D. He, Y.H. Tang, and X.L. Wang. Dissipative particle dynamics simulation on the membrane formation of polymer-diluent system via thermally induced phase separation. *J. Memb. Sci.*, 368:78–85, 2011.

[9] Y.H. Tang, Y.D. He, and X.L. Wang. Effect of adding a second diluent on the membrane formation of polymer/diluent system via thermally induced phase separation: Dissipative particle dynamics simulation and its experimental verification. *J. Memb. Sci.*, 409-410:164–172, 2012.

[10] Y.H. Tang, Y.D. He, and X.L. Wang. Three-dimensional analysis of membrane formation via thermally induced phase separation by dissipative particle dynamics simulation. *J. Membr. Sci.*, 437:40–48, 2013.

[11] Y.H. Tang, Y.D. He, and X.L. Wang. Investigation on the membrane formation process of polymer-diluent system via thermally induced phase separation accompanied with mass transfer across the interface. dissipative particle dynamics simulation and its experimental verification. *J. Memb. Sci.*, 474:196–206, 2015.

[12] Y.H. Tang, H.H. Lin, T.Y. Liu, H. Matsuyama, and X.L. Wang. Multiscale simulation on the membrane formation process via thermally induced phase separation accompanied with heat transfer. *J. Memb. Sci*, 515:258–267, 2016.

[13] Y. Termonia. Monte carlo diffusion model of polymer coagulation. *Phys. Rev. Lett.*, 72: 3678–3681, 1994.

[14] Y. Termonia. Fundamentals of polymer coagulation. *J. Polym. Sci. B Polym. Phys.*, 33:279–288, 1995.

[15] Y. Termonia. Molecular modeling of phase-inversion membranes: effect of additives in the coagulant. *Journal of Membrane Science*, 104:173–179, 1995.

[16] X. He, C. Chen, Z. Jiang, and Y. Su. Computer simulation of formation of polymeric ultrafiltration membrane via immersion precipitation. *J. Memb. Sci.*, 371:108–116, 2011.

[17] G.T. Caneba and D.S. Soong. Polymer membrane formation through the thermal-inversion process. 2. mathematical modeling of membrane structure formation. *Macromolecules*, 18:2545–2555, 1985.

[18] P.K. Chan and A.D. Rey. Computational analysis of spinodal decomposition dynamics in polymer solutions. *Macromol. Theory Simul.*, 4:873–899, 1995.

[19] B. Barton, P.Graham, and A. McHugh. Dynamics of spinodal decomposition in polymer solutions near a glass transition. *Macromolecules*, 31:1672–1679, 1998.

[20] K. W D. Lee, P.K. Chan, and X. Feng. A computational study into thermally induced phase separation in polymer solutions under a temperature gradient. *Macromol. Theory Simul.*, 11:996–1005, 2002.

[21] K.W.D. Lee, P.K. Chan, and X. Feng. Morphology development and characterization of the phase-separated structures resulting from the thermal-induced phase separation phenomenon in polymer solutions under a temperature gradient. *Chem. Eng. Sci.*, 59: 1491–1504, 2004.

[22] S.B. Kukadiya, P.K. Chan, and M. Mehrvar. The ludwig-soret effect on the thermally induced phase separation process in polymer solutions: A computational study. *Macromol. Theory Simul.*, 18:97–107, 2009.

[23] Y. Mino, T. Ishigami, Y. Kagawa, and H. Matsuyama. Three-dimensional phase-field simulations of membrane porous structure formation by thermally induced phase separation in polymer solutions. *J. Membr. Sci*, 483:104–111, 2015.

[24] A.J. McHugh B.F. Barton. Kinetics of thermally induced phase separation in ternary polymer solutions. i. modeling of phase separation dynamics. *Journal of Polymer Science Part B: Polymer Physics*, 37:1449–1460, 1999.

[25] R. Saxena and G.T. Caneba. Studies of spinodal decomposition in a ternary polymer-solvent-nonsolvent system. *Polymer Engineering and Science*, 42:1019–1031, 2002.

[26] B. Zhou and A.C. Powell. Phase field simulations of early stage structure formation during imersion precipitation of polymeric membranes in 2d and 3d. *J. Membr. Sci.*, 268:150–164, 2006.

[27] Manuel Hopp-Hirschler and Ulrich Nieken. Modeling of pore formation in phase inversion processes: Model and numerical results. *Journal of Membrane Science*, 564:820 – 831, 2018.

[28] A. Akthakul, C.E. Scott, A.M. Mayes, and A.J. Wagner. Lattice boltzmann simulations of asymmetric membrane formation by immersion precipitation. *J. Membr. Sci.*, 249: 213–226, 2005.

[29] D.R. Tree, K.T. Delaney, H.D. Ceniceros, T. Iwama, and G.H. Fredrickson. A multi-fluid model for microstructure formation in polymer membranes. *Soft Matter*, 13:3013–3030, 2017.

[30] D.R. Tree, T. Iwama, K.T. Delaney, J. Lee, and G.H. Fredrickson. Marangoni flows during nonsolvent induced phase separation. *ACS Macro Lett.*, 7:582–586, 2018.

[31] Y. Lin, Y. Tang, H. Ma, J. Yang, Y. Tian, W. Ma, and X. Wang. Formation of a bicontinuous structure membrane of polyvinylidene fluoride in diphenyl carbonate diluent via thermally induced phase separation. *J. Appl. Polym. Sci.*, pages 1523–1528, 2009.

[32] Lishun Wu and Junfen Sun. Structure and properties of pvdf membrane with pes-c addition via thermally induced phase separation process. *Applied Surface Science*, 322: 101 – 110, 2014.

[33] Min Zhang and Yun-Ren Qiu. Effects of coagulation bath temperature on structure and performance of poly (vinyl butyral) hollow fiber membranes via thermally induced phase separation. *J. Cent. South Univ.*, 21:3057–3062, 2014.

[34] G.D.J. Phillies. Universal scaling equation fro self-diffusion by macromolecules in solution. *Macromolecules*, 19:2367–2376, 1986.

[35] G.D.J. Phillies. The hydrodynamic scaling model for polymer self-diffusion. *J. Phys. Chem*, 93:5029–5039, 1989.

[36] Poly(vinylidene fluoride), January 2016. URL `https://polymerdatabase.com/polymers/polyvinylidenefluoride.html`.

[37] W. K. lewis. The evaporation of a liquid into a gas. *Transactions of the American Society of Mechanical Engineers*, pages 325–340, 1922.

[38] S. Frijters, T. Kruger, and J. Harting. Parallelised hoshen-kopelman algorithm for lattice-boltzmann simulations. *Comput. Phys. Commun.*, 189:92, 2015.

[39] M.B. Wise and P.C. Millett. Two-dimensional bicontinuous structures from symmetric surface-directed spinodal decomposition in thin films. *Phys. Rev. E.*, 98:022601, 2018.

[40] B. Ladewig and M.N.Z. Al-Shaeli. *Fundamentals of Membrane Bioreactors.* Springer Nature Singapore Pte Ltd., 2017.

[41] Sungil Jeon, Hamed Karkhanechi, Li-Feng Fang, Liang Cheng, Takahiro Ono, Ryota Nakamura, and Hideto Matsuyama. Novel preparation and fundamental characterization of polyamide 6 self-supporting hollow fiber membranes via thermally induced phase separation (tips). *Journal of Membrane Science*, 546:1 – 14, 2018.

Chapter 5

Paper 2: Phase-field modeling of non-solvent induced phase separation (NIPS) for PES/NMP/Water with comparison to experiments

5.1 Abstract

We develop a phase-field model to simulate the formation of porous polymeric membranes via non-solvent induced phase separation. The material system of interest is PES/NMP/Water (Polyethersulfone/N-methyl-2-pyrrolidone/Water), however the approach is broadly applicable to other materials. The three-component system is represented with two field variables: one representing the volume fraction of polymer, and the other the fractional composition of non-solvent N (water) vs solvent S (NMP). The exchange of solvent and non-solvent is solved with a Fickian diffusion model, thus capturing the in-flux of the coagulation bath into the polymer solution. As a demonstration of the predictive capabilities of the model, the concentration of solvent (NMP) in the coagulation bath was varied to draw comparisons with experiments. Two- and three-dimensional simulations were carried out to evaluate the cross-sectional pore morphology and the top surface pore size for membranes formed by NIPS. Experiments involving handcast membranes of a similar system were performed for comparison with the simulations, and an agreement was found concerning the dependence of pore morphology on the composition of the coagulation bath.

5.2 Introduction

One of the primary methods for fabricating flat-sheet membrane filters involves non-solvent induced phase separation (NIPS), whereby a polymer solution is exposed to a coagulation bath containing a non-solvent (otherwise known as a poor solvent) and the polymer precipitates out of solution resulting in a porous network [1–4]. A depiction of the NIPS process is shown in Fig. 5.1. The mechanisms pertaining to the formation of specific morphologies and defects are complex and mostly explained by heuristic knowledge of experimental process conditions.

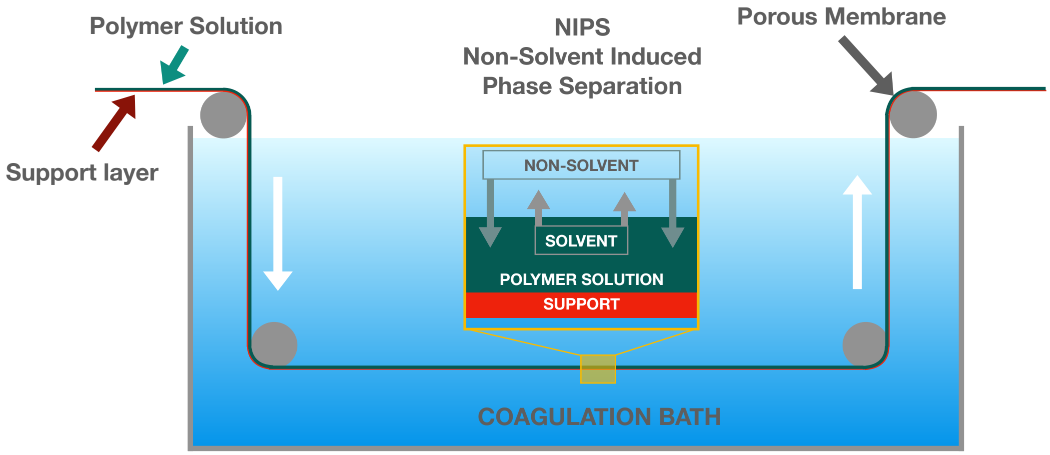

Figure 5.1: Schematic of the NIPS membrane formation process.

Computational modeling of the membrane formation process is a developing branch of research with the aim to accelerate industrial R&D, aid in tailoring membrane performance, and assist in developing new membranes from novel materials. The large number of process control variables, cost of experimental exploration, and time-consuming research and development are primary motivations for computational research in an effort to understanding *in-situ* phenomenon and predict membrane morphologies. Different modeling techniques have been applied to this specific problem and include molecular-scale simulations, mesoscale simulations, and macroscale simulations.

Molecular-scale simulations are capable of predicting polymer chain behavior during phase inversion; however, the length scales are in the range of tens of nanometers which is at least one order of magnitude smaller than a full membrane thickness. Dissipative particle dynamics (DPD) have been widely utilized and can predict solution behavior during phase separation at nanoscale dimensions (up to 100-200 nm). Work done by Wang et al. [5] investigated the basics of NIPS in two dimensions by observing the exchange of non-solvent (into the polymer solution) and solvent (out of the polymer solution) and the resulting phase separation process. The kinetics of phase inversion have also been characterized with DPD simulations with relationships drawn to the mesoscale [6]. The effect of additives on membrane morphology studied by Tang et al. [7] show how the strength of interactions for the additive and other components can have a significant effect on membrane structure. Further work by Tang et al. [8–10] extended the previous work into three dimensions and incorporated mass transfer at the coagulation bath interface with a focus on the formation of a dense pore layer at this interface. This work has been expanded upon to understand how the strength of interaction between non-solvent and solvent affect the rate of phase separation and the resulting morphology [11].

Modeling work that has been carried out in the mesoscale using the phase-field (PF) method is proving capable of capturing pore morphologies and domain sizes closer to the thickness of a filtration membrane. One-dimensional simulations by Caneba et al. [12] used the Cahn-Hilliard equation with the Flory-Huggins (FH) free energy of mixing to investigate the formation of anisotropic membrane structures. Two-dimensional simulations by Barton et al. [13, 14] investigated the thermodynamic and transport properties of phase separation. The principal findings show that increasing polymer concentration slows the diffusive process of phase inversion during thermally induced phase separation (TIPS).Two dimensional simulations carried out by Fernandes et al. [15] explored a simplified model in order to reduce the number of parameters needed for simulations. The principal findings from the study include morphological changes to the dense pore region when adding solvent to the coagulation bath, adding non-solvent to the polymer solution, and the effect of membrane thickness on the initial casting solution.

Two and three-dimensional simulations coupled to the Navier-Stokes equation by Zhou et al. [16] also found the development of a dense pore region and characterized the effect of polymer concentration in the casting solution. Hopp-Hirschler and Nieken [17] conducted two-dimensional simulations with an imposed moving precipitation front, the velocity of which affected the morphology of pores. Two and three dimensional simulations carried out by Tree et al. [18–20] explored coarsening kinetics, Marangoni flows, and the inclusion of mass transfer and their effects on final morphology. Three dimensional TIPS simulations carried out by Mino et al. [21] looked at the late-stage morphological development of spinodal decomposition and how it was affected by early-stage morphology. Cervellere et al. [22] conducted three dimensional simulations of TIPS and found that the coagulation bath temperature and polymer concentration has a large effect on the depth of the dense pore region and the overall pore size, with higher coagulation bath temperatures and lower polymer concentrations favoring larger pores.

The Lattice-Boltzmann method has also been used recently to study the formation of anisotropic membrane structures [23]. Work done by Gan et al. [24, 25] evaluated the role of component viscosities during TIPS, showing that the velocities of local flow were found to be inversely proportional to temperature, exemplifying the connection to casting temperature and morphology. Further work done by Gan et al. [26] included surface tension effects for the system thermodynamics and found two domains of phase separation - spinodal decomposition and nucleation/growth - however more work needs to be done to fully understand the coarsening kinetics.

The NIPS process has been studied by computational researchers in efforts to verify various models with experimental observations; however, a knowledge gap still exists relating to the formation of a dense skin layer as well as the origin of macrovoid formation. The Monte Carlo method was used by He et al. [27] and found that the diffusion of non-solvent from the coagulation bath is exponentially decreased by the presence of polymer. It was also found that spinodal decomposition and nucleation/growth both occurred at different depths within the simulated membrane structure. The work done by Tree et al. [19] (mentioned previously) examined the effect of Marangoni flows on membrane formation however the results were inconclusive and showed the need for a method of implementing polymer vitrification, which was not included.

In this paper, we utilize a PF model to conduct both 2D and 3D simulations of the NIPS process in the specific material system of PES/NMP/water. Here, we specifically investigate the effect of independently varying two key parameters: the polymer volume fraction in solution and the coagulation bath composition. In addition, a model to capture vitrification of the polymer-rich phase has been implemented and found to satisfactorily halt coarsening in regions with high polymer content. Combining this mesoscale model with modern high-performance computing enables large simulation domains in the micrometer scale that allow direct comparisons with scanning-electron microscope images. The NIPS simulations are evaluated by observing the cross-sectional and top-surface morphology given different casting conditions. The simulations are then compared to handcast membranes produced with similar casting conditions.

5.3 Methods

Our PF model evolves two field variables that together sufficiently describe the three-component systems typical of NIPS processing. The first field variable represents the polymer volume fraction, ϕ_P. The two remaining components (water and NMP) are both small-molecule solvents that are fully miscible with one another. Therefore, we treat the solvent as a two-component mixture, and the compositional fraction of which is represented by f_N whereby NS signifies non-solvent which in this case corresponds to water. As there are exactly two solvents, one can tacitly determine the amount of NMP as $1 - f_N$. The polymer volume fraction is evolved with a Cahn-Hilliard-Cook equation [28]:

$$\frac{\partial \phi_P}{\partial t} = \nabla \cdot \left(M_P \nabla \left(\frac{\delta F_{mix}}{\delta \phi_P} - 2\kappa \nabla^2 \phi_P \right) \right) + \xi, \tag{5.1}$$

where M_P is the polymer mobility, F_{mix} is the free energy of mixing, κ scales the interfacial energy between the two phases, and ξ is a random number centered at zero that imparts a small thermal fluctuation to the polymer concentration. The polymer mobility is related to the polymer diffusivity by the equation [16]:

$$M_P = \frac{D_P}{\partial^2 F_{mix} / \partial \phi_P^2} \tag{5.2}$$

where D_P is the diffusion coefficient of a polymer chain in solution.

The binary FH free energy of mixing used in this work is given below [29]:

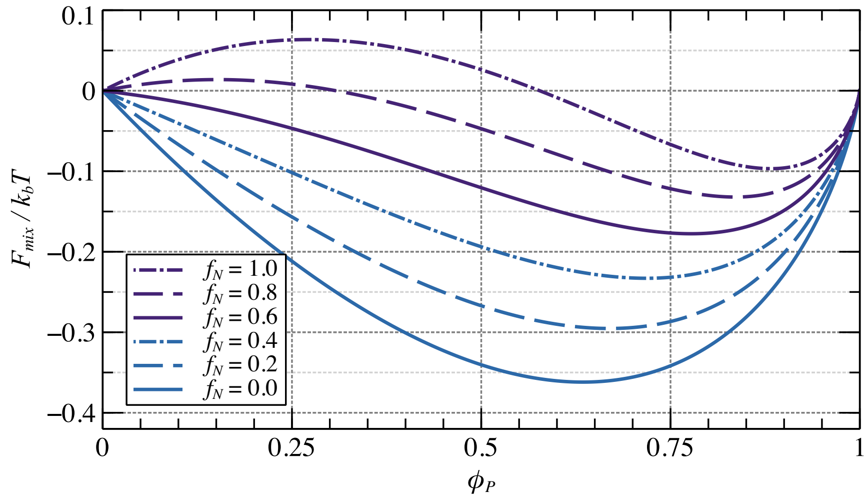

Figure 5.2: Free energy of mixing for six different non-solvent fractions f_N calculated from Eqs. 5.3 and 5.4. As f_N increases, the polymer solution goes from miscible to immiscible.

$$F_{mix} = k_b T \left[\frac{\phi_P}{N} \ln \phi_P + \phi_S \ln \phi_S + \chi \phi_P \phi_S \right], \tag{5.3}$$

where the substitution $\phi_S = 1 - \phi_P$ is made for the solvent. The degree of polymerization is $N = 150$, the temperature is 298 K (25°C), and χ is the interaction parameter between polymer and solvent. This interaction parameter determines miscibility between polymer and solvent and can be dependent on temperature, composition, or a combination of process conditions that are prevalent during casting. In this work, we assume isothermal conditions and therefore χ depends solely on composition with the below weighted average:

$$\chi = f_N \chi_{PN} + (1 - f_N)\, \chi_{PS}, \tag{5.4}$$

where $\chi_{PN} = 1.5$ is the interaction parameter between PES and water and $\chi_{PS} = 0.034$ is the interaction parameter between PES and NMP, both values set for $T = 25$°C [30–32]. As the local non-solvent fraction increases, χ also increases and can ultimately surpass the critical value that in turn activates phase separation. The FH free energy of mixing is directly affected by the presence of non-solvent as depicted Fig. 5.2 where the FH free energy is plotted for a range of f_N.

We assume that the exchange of solvents between the polymer solution and the coagulation bath (water into and NMP out of the polymer solution) is a diffusion-governed process, and we use a Fickian diffusion model to evolve f_N. Therefore, in the current implementation, we ignore hydrodynamic transport, although that can be added in the future. The evolution equation is written as:

$$\frac{\partial f_N}{\partial t} = \nabla \cdot \left(D_N(\phi_P)\nabla f_N\right), \tag{5.5}$$

where D_N represents the diffusivity of the non-solvent species, and it is dependent of the local value of ϕ_P hence it is spatially heterogeneous.

The diffusivity of polymer in solution is described with the Phillies model [33, 34]:

$$D_P = D_P^o exp(-\alpha c^\nu) \tag{5.6}$$

where the pre-exponential D_P^o is the diffusion coefficient of a single polymer chain in solution (the dilute limit), α and ν are scaling coefficients to fit experimental data, and c is the mass concentration of polymer with units of g/L. This mass concentration is calculated from ϕ_P according to $c = \phi_P(M_w/M_{vol})$, where M_w = 232.36 g/mol and M_{vol} = 0.1683 L/mol for PES. The scaling coefficients are assigned values of α = 0.1 and ν = 0.6 to achieve a similarly shaped diffusion curve that has been reported for PES in NMP [30]. After converting diffusivity to mobility (Eq. 5.2) the mobility was scaled by a factor of 0.35 to achieve the same magnitude as the mobility derived from experimental diffusivity data previously reported [30].

Equation 5.6 can describe general concentration-dependent diffusion and in this work was also used to calculate D_N:

$$D_N = D_N^o exp(-\alpha c^\nu) \tag{5.7}$$

where the pre-exponential D_N^o is the diffusion coefficient of a water molecule in the absence

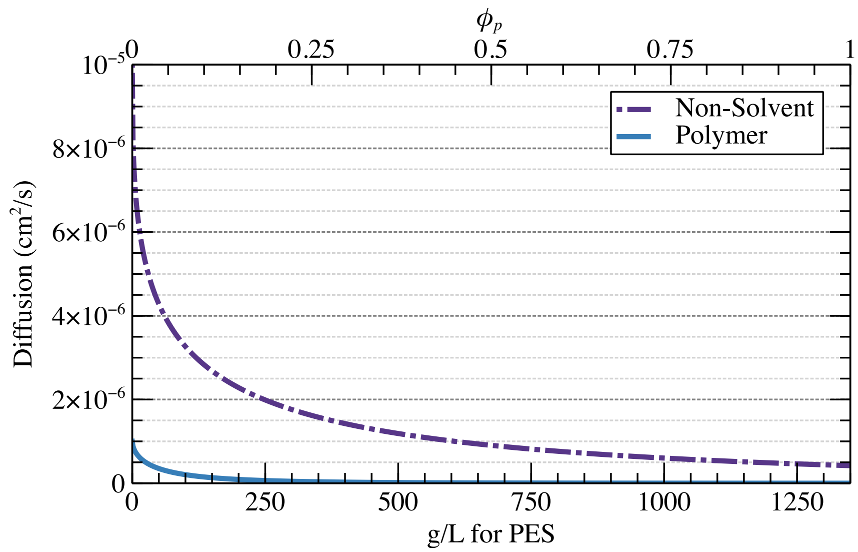

Figure 5.3: Diffusivities for both polymer (PES) and water (non-solvent) calculated from Eq. 5.6. The diffusivities for both species decrease with increasing polymer content. The top x-axis represents the non-solvent concentration and the bottom x-axis represents the PES concentration.

of polymer. The scaling coefficients for non-solvent diffusion were $\alpha = 0.2$ and $\nu = 0.4$ which allow for more rapid diffusion of non-solvent while reducing the concentration dependence for D_N in comparison to D_P [30]. The value of D_N^o was set an order of magnitude higher than that of D_P^o accounting for more rapid diffusion of water molecules as compared to polymer chains. Here, $D_N^o = 10^{-5}$ cm^2/s and $D_P^o = 10^{-6}$ cm^2/s which is similar to what has been reported in the literature [30, 35]. The diffusion curves for polymer and non-solvent are shown in Fig. 5.3.

A cutoff volume fraction, ϕ_{cutoff}, was used to vitrify the polymer domain in effect freezing the polymer-rich structures after phase separation. Once the polymer volume fraction exceeds a threshold, $\phi_P \geq \phi_{cutoff}$, the mobility is drastically reduced by dividing the mobility by a factor of 10^6, vitrifying the polymer domain. This study uses a cutoff volume fraction of $\phi_{cutoff} = 0.75$ similar to polymer vitrification concentrations experimentally observed by Kim et al. [36]. All simulations below were run until complete vitrification occurred throughout the simulation domain.

To execute the simulations, Eqs. (5.1) and (5.5) are solved with an explicit finite difference scheme on a rectilinear grid with uniform spacing. To reduce round-off error, the equations are solved in reduced units of length ($\bar{l}$) and time ($\bar{t}$). The grid spacing $\Delta x = 1\ \bar{l}$, the time step size is $\Delta t = 0.01\ \bar{t}$, and D_P^o is set to unity (and hence D_N^o is set to 10). Following the simulations, the time and length scales are converted into physical units by assuming $\bar{l} = 35$ nm (i.e. each grid node represents a box with side lengths of 35 nm) and $\bar{t} = \bar{l}^2/\left(10^{-6}\text{cm}^2/\text{s}\right) = 1.225\times10^{-5}$ms. These values were chosen to allow for a simulation time window of 10 seconds and pore sizes in the micro-filtration range. In Eq. (5.1), the thermal fluctuation term ξ is a random number uniformly chosen within the bounds [-0.05,0.05], and the interfacial energy term κ is set to unity. Setting κ to unity in conjunction with grid spacing $\Delta x = 1\ \bar{l}$ resolves the interface between two phases over 3-7 grid points.

The simulation domain has periodic boundary conditions imposed in the x- and y-directions with no-flux boundary conditions in the z-direction. Two-dimensional simulations exist in the xz-plane. The simulations are initialized with small fluctuations in ϕ_P about the mean value $\bar{\phi}_p$, and $f_N = 0$ inside the domain. The top surface in the z-direction represents the interface between the polymer solution and the coagulation bath. On this top surface, we assign a fixed value of f_N, which we denote as f_N^{CB}, that is constant in time and imposes a time-dependent gradient of non-solvent within the domain thereby driving non-solvent diffusion into the polymer solution. This study looks at the effect of adding NMP to the coagulation bath, and we consider three different coagulation bath compositions $f_N^{CB} = 1.0$, 0.8, and 0.6, which correspond to 100% water, 80% water/ 20% NMP, and 60% water/40% NMP, respectively. Figure 5.4 illustrates the co-evolution of ϕ_P and f_N in a two-dimensional simulation of the NIPS process. Although it is not shown, we have observed that the presence of the polymer-rich domains significantly slows down the in-flux of water due to the reduction of D_N in those regions. See Supplemental Movie 1 for an animation of Fig. 5.4.

Handcast membranes were prepared for morphological comparison to simulations. The recipe for the membrane was 15 wt% BASF PES E3010, 10 wt% PVP k90 (to increase solution viscosity and suppress macrovoids) and NMP. The polymer solution was mixed at

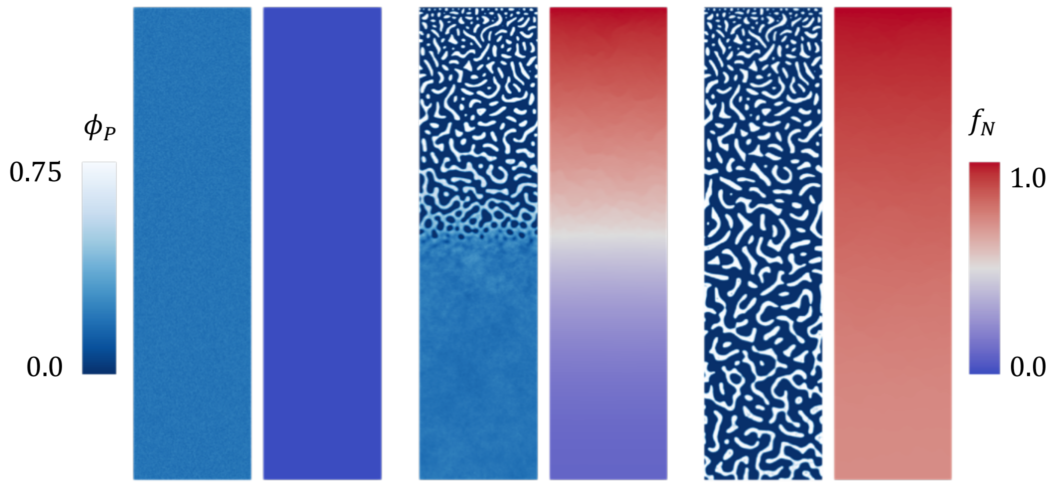

Figure 5.4: Snapshots of a 2D simulation at times 0s (left), 0.735s (middle), and 2.205s (right). The fields of ϕ_P and f_N for each time are shown side-by-side. The top surface maintains a constant value of $f_N^{CB} = 1.0$, and as non-solvent diffuses into the domain, phase separation occurs from top to bottom. Here, the average polymer volume fraction is $\bar{\phi}_P = 0.2$. The vitrification model freezes polymer-rich domains when $\phi_P > 0.75$.

50 °C for at least 24 hours. The polymer solution was then cast onto a plastic film taped to a glass pane heated to a temperature of 50 °C. The casted film was then inserted into the coagulation bath which was held at 50 °C and consisted of DI water and different vol% NMP (0%, 20%, and 40%). The membrane was then removed from the coagulation bath and soaked in DI water for 24 hours before drying. The scale of the simulations and the region of interest for cross section comparisons are detailed in Fig. 5.5, where the top 40 μm above the macrovoids are considered.

Current simulation methods are unable to capture the formation of macrovoids and accordingly the model presented simulates the formation of an idealized membrane structure free of defects. Membranes made with a polymer solution containing only PES and NMP have a large amount of macrovoids initiating at the top surface of the membrane and the addition of PVP helps to suppress the formation of macrovoids without drastically affecting the thermodynamics of mixing [37, 38]. The addition of PVP also increases solution viscosity which in turn decreases the rate of solvent-nonsolvent exchange occurring at the interface of the coagulation bath and the polymer solution. This change moves the compo-

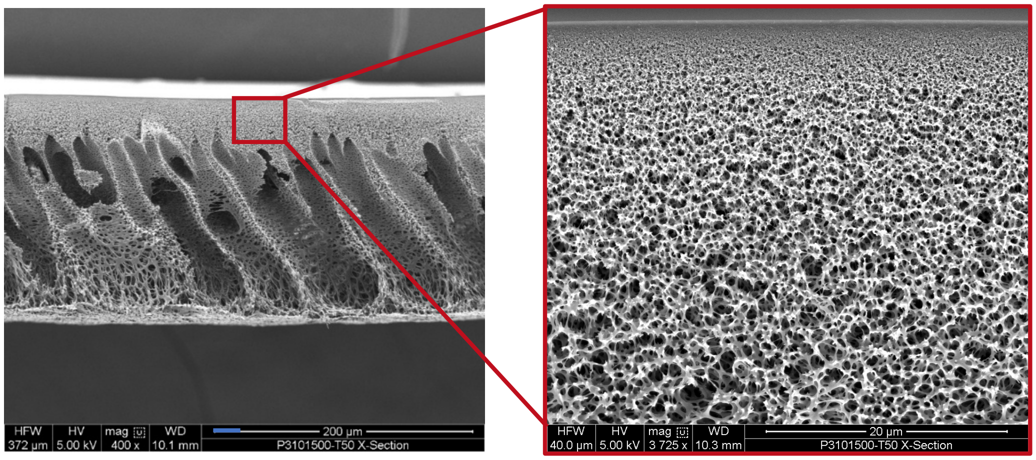

Figure 5.5: Two SEM images showing the approximate scale of the simulations for drawing comparisons. The left image is the whole membrane cross section and the right image is a closer view corresponding to the scale of the simulations. The scale bar on the left image is 200 μm and the scale bar on the right image is 20 μm.

sition path from instantaneous to delayed demixing which results in a morphological change from fingerlike macrovoids to a dense bicontinuous structure [3, 39].

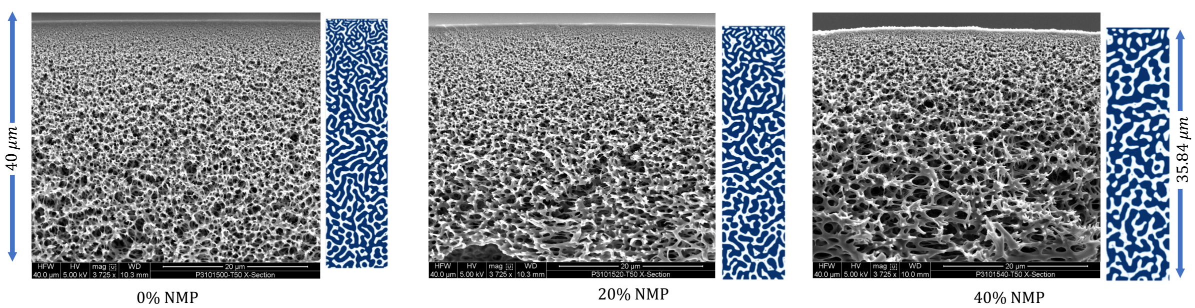

Figure 5.6: Cross-sectional comparison between handcast membranes and simulations. From left to right the coagulation bath contained 0, 20, and 40 vol% NMP ($f_N^{CB} = 1.0$, 0.8 and 0.6). The simulations show good qualitative agreement with experiments, both exhibiting anisotropic pore structures that increases in size as NMP is added to the coagulation bath. Here, the simulations have a $\bar{\phi}_P = 0.25$.

5.4 Results & Discussion

We performed simulations in both two and three dimensions to evaluate the effect of coagulation bath composition and polymer volume fraction on pore morphology. The two-dimensional simulations are clearly less computationally demanding, and this allowed us to extend the domain to greater depths thereby enabling better insight into the development of anisotropic pore structures. On the other hand, our 3D simulations were conducted with relatively shallower depths, but provided analysis of the top-surface pore morphology that is critical for membrane performance.

The domain size for the 2D simulations is 8.96×35.84 μm (resolved with a 256×1024 grid). Hence, the depths of our 2D simulations closely correspond with the experimental membrane depth shown on the right side of Fig. 5.5. The domain size for the 3D simulations is $8.96 \times 8.96 \times 2.24$ μm (resolved with a $256 \times 256 \times 64$ grid). We considered three different coagulation bath compositions: $f_N^{CB} = 1.0$, 0.8, and 0.6, which represent adding NMP into the coagulation bath at 0 vol%, 20 vol%, and 40 vol%, respectively. In addition, three different polymer volume fractions were simulated: $\bar{\phi}_P = 0.2$, 0.225, and 0.25. Hence, nine unique test cases were simulated in both 2D and 3D domains.

Our criterion for terminating a simulation was full vitrification throughout the domain. This criterion was met at different times for the three different coagulation bath compositions due to the fact that the f_N field evolves faster when its value at the top surface is higher. The two-dimensional simulations required 4.9s, 7.35s, and 9.8s to reach full vitrification for the three coagulation bath compositions $f_N^{CB} = 1.0$, 0.8 and 0.6, respectively.

A comparison between the 2D simulations with a $\bar{\phi}_P = 0.25$ and handcast membrane cross sections are seen in Fig. 5.6. The 2D simulations exhibit a dense pore region near the top surface where the non-solvent is in contact with the polymer solution. These initial pores develop quickly, as does vitrification, due to the early influx of non-solvent into the polymer solution. At greater depths, whereby the in-flux of non-solvent is slower, larger pores are able to develop due to a slower nucleation process as well as a certain degree of phase coarsening that occurs. The addition of NMP to the coagulation bath (decreasing f_N^{CB} from 1.0 to 0.8 and 0.6) has a notable effect on morphology. When there is no NMP in the coagulation bath ($f_N^{CB} = 1.0$) very small pores rapidly form near the top surface of the domain resulting in a dense pore region. When 20 vol% NMP is added to the coagulation bath, the dense pore region near the top surface decreases in thickness and the entire domain exhibits larger

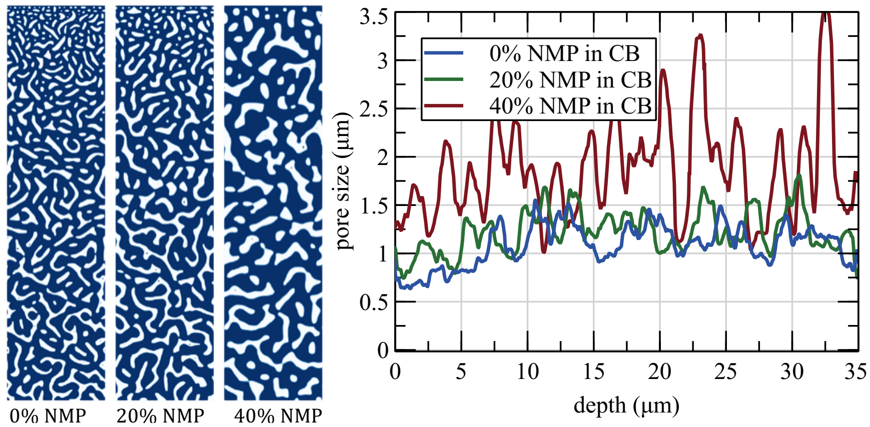

Figure 5.7: (left) Final structures from three different 2D simulations with different coagulation bath compositions (as indicated), each with $\bar{\phi}_P = 0.2$. (right) Pore size versus membrane depth for the three coagulation bath compositions. As NMP is added to the coagulation bath the pore size at the top surface increases along with the the pore size throughout the membrane structure.

pores. When 40 vol% NMP is added to the coagulation bath the most drastic difference is seen with the reduction in the dense pore region thickness and larger overall pore size.

In Fig. 5.7, we show a side-by-side comparison of the 2D pore morphologies for $\bar{\phi}_P = 0.2$, as well as a plot of the pore size versus membrane depth. In order to evaluate the pore size for the 2D simulations, each layer from top to bottom was scanned left to right and the average widths of the polymer-poor domains were computed (this was done via tracking the distance between phase interfaces). These widths were recorded as the pore size for that layer. To reduce noise, the pore depths were averaged within bins consisting of 31 layers, which are shown in Fig. 5.7. The plot quantitatively demonstrates that a higher amount of NMP in the coagulation bath results in a larger pore size. This trend is also seen with $\bar{\phi}_P = 0.225$ and $\bar{\phi}_P = 0.25$, as shown in Figs. 5.8 and 5.9, respectively. In addition, it is found that the pore sizes decrease with increasing $\bar{\phi}_P$, which is expected. The porosity of the membranes at the end of the simulations correlates to the volume fraction of the polymer-rich phase, which can be calculated using a lever rule with a tie line extending from $\phi_P = 0.0$ to $\phi_P = \phi_{cutoff} = 0.75$. The porosity for $\bar{\phi}_P = 0.2$, 0.225, and 0.25 is 0.73, 0.70, and 0.67 respectively.

The 3D simulations, with the dimensions given at the beginning of this section, were

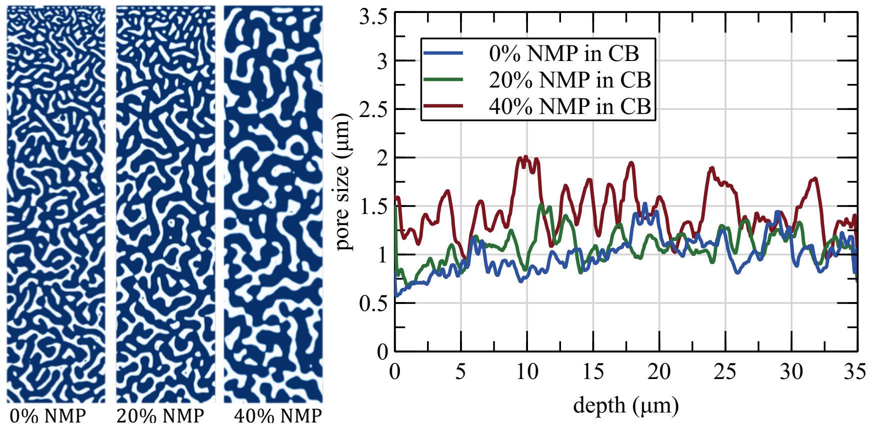

Figure 5.8: Same as Fig. 5.7, but with a polymer volume fraction of $\bar{\phi}_P = 0.225$.

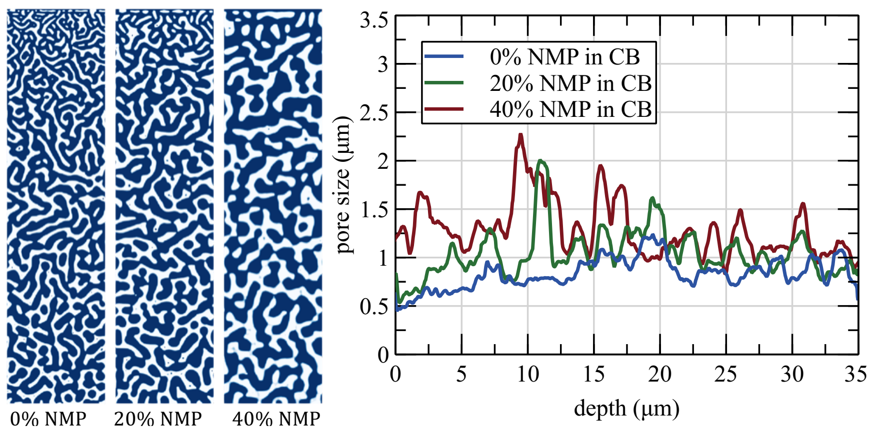

Figure 5.9: Same as Fig. 5.7, but with a polymer volume fraction of $\bar{\phi}_P = 0.25$.

executed with the same conditions used in the 2D simulations described above. Due to the smaller depth, the 3D simulations required less physical time to reach full vitrification. The simulations were run for 0.1225s, 0.3575s, and 1.225s for coagulation bath compositions of $f_N^{CB} = 1.0$, 0.8, and 0.6 (0 vol% NMP, 20 vol% NMP, and 40 vol% NMP), respectively. The motivation for the 3D simulations was to analyze the pore morphology on the top surface of the membrane, and how it depends on the physical conditions we varied. Supplemental Movies 2 - 5 provide animations of the 3D simulations.

Figure 5.10 shows both top-down and cross-sectional views of the pore morphology for the nine unique conditions tested. The general trends found in the 2D simulations carry over to the 3D simulations, namely that pore size generally increases with increasing NMP

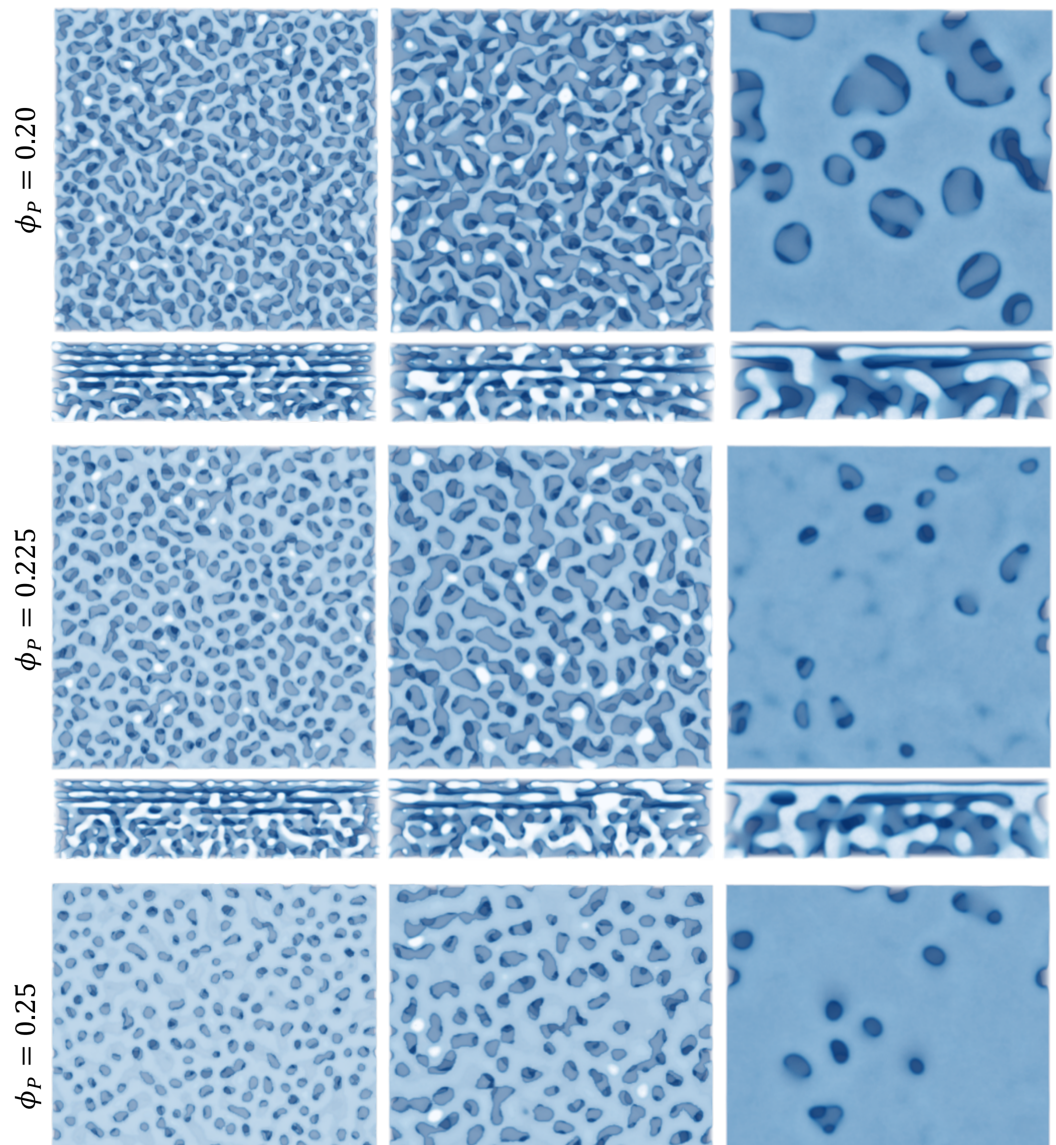

Figure 5.10: 3D simulations with top-down and cross-sectional views illustrating the dependence of pore structure at the top surface with varying coagulation bath composition and polymer volume fraction. As NMP is added to the coagulation bath, the pore size at the top surface increases for each $\bar{\phi}_P$ represented. As $\bar{\phi}_P$ increases the pores decrease in size and the dense pore region fills in with more polymer.

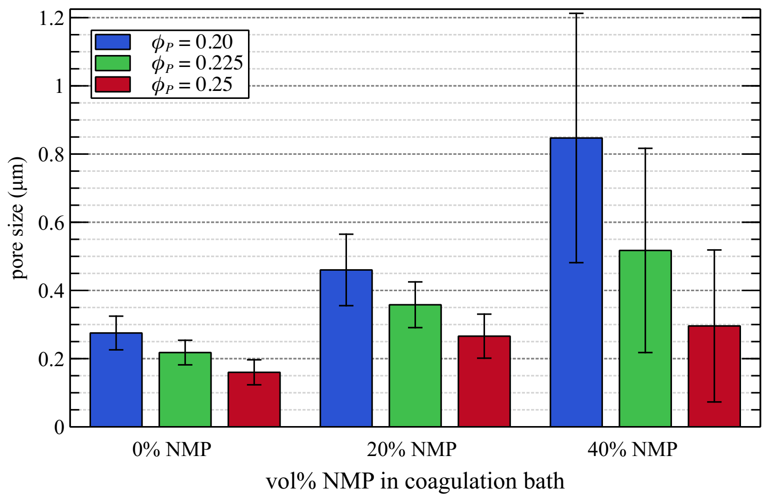

Figure 5.11: Pore size at the membrane top surface versus coagulation bath composition for the three different polymer volume fractions. As NMP is added to the coagulation bath the pore size at the surface increases for each $\bar{\phi}_P$ represented. As $\bar{\phi}_P$ increases the pores decrease in size.

content in the coagulation bath and decreasing polymer volume fraction. However, the 3D simulations also reveal distinct characteristics of the pore morphology at the top surface. In particular, we observe pore structures on the top surface that are continuous in some cases and discrete in other cases. For example, with $\bar{\phi}_P = 0.2$, the pore structure is continuous along the top surface with 0% NMP in the coagulation bath, but transitions to a discrete pore structure with 40% NMP in the coagulation bath. In addition, when the coagulation bath composition is fixed at 0% NMP, the pore structure changes from continuous to discrete when the polymer volume fraction increases from $\bar{\phi}_P = 0.2$ to $\bar{\phi}_P = 0.25$.

The quantitative values for pore size on the top surface from the 3D simulations are shown in Fig. 5.11. These pore size values were computed by scanning along two planes within the top layer of the membrane structure. The calculation is similar to that for the 2D simulations, however for each xy-plane a scan was conducted along the x-direction for each y-value. This provided more data, for which the average and the standard deviation (error bars) are shown in Fig. 5.11. Again, consistent with the 2D results, the pore size increases for all $\bar{\phi}_P$ when NMP is added to the coagulation bath.

A comparison of the top-surface pore morphology between the experiments and the 3D

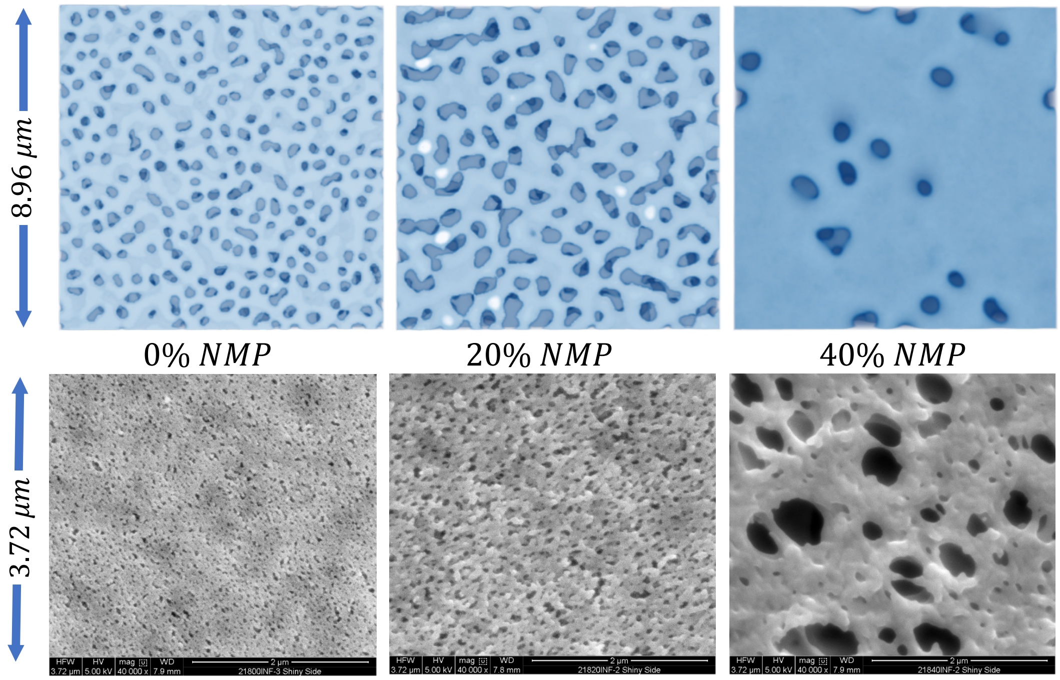

Figure 5.12: Comparison of top surface pore morphology between experiments and simulations. From left to right the coagulation bath contained 0, 20, and 40 vol% NMP (f_N^{CB} = 1.0, 0.8 and 0.6). The simulations here have a polymer volume fraction of $\bar{\phi}_P = 0.25$.

simulations is given in Fig. 5.12. The pore size increases as NMP is added to the coagulation bath for both the experiments and simulations. The most drastic difference in morphology occurs when the coagulation bath contains 40 vol% NMP. In general, we find that the pore sizes for the top surface are larger in the simulations by approximately a factor of two to four when compared with experiments (see Figs. 5.11 and 5.12). The model does not accurately capture the smallest pores that form in the dense pore region near the top surface of the membranes as seen Fig. 5.6. We attribute this discrepancy to the length- and time-scale resolution of the simulations. Initial phase separation at the top surface occurs rapidly and would require a smaller grid spacing and time step to resolve the smaller pore sizes present in the experiments. The simulations in this work provide guidance for future studies by identifying the process conditions that warrant higher-resolution PF simulations to better study the top-surface morphology.

5.5 Conclusion

The model presented here yields qualitative and quantitative insight to how different casting parameters can affect the pore size, anisotropic pore morphology, and surface pore size for membrane filters casted with PES/NMP/ Water. Two-dimensional simulations show that adding NMP to the coagulation bath decreased the thickness of the dense pore region near the top surface and increased the overall pore size throughout the cross section. Three-dimensional simulations show a significant variability of pore size in the top membrane surface with surface pores increasing in size with the addition of NMP into the coagulation bath. The top surface also exhibited a transition from continuous to discrete morphology as $\bar{\phi}_P$ increased from 0.2 to 0.25. As expected both two- and three-dimensional simulations show that pore size decreases as polymer concentration increases. Comparisons drawn from handcast membranes showed similar trends, namely larger surface pores and a decrease in the dense pore region near the top surface when NMP is added to the coagulation bath. The pore sizes in the dense pore region near the top surface observed in experiments are smaller in size compared to the simulations. This discrepancy in size is attributed to the rapid formation of these regions where the coagulation bath comes into contact with the polymer solution. To fully resolve these differences a reduction in the time step size and grid spacing is required, which we leave for future work.

5.6 Acknowledgements

The authors would like to acknowledge funding from the MAST Center (Membrane Research Science and Technology Center) through NSF. The authors would like to acknowledge funding from the INTERN program in collaboration with MilliporeSigma. The authors would also like to acknowledge the Arkansas High Performance Computing Center for the computational resources. The authors would like to acknowledge Susan Connolly and David Bell at MilliporeSigma for imaging membrane samples. The authors would like to acknowledge mentors from MilliporeSigma, 3M, and the MAST Center for guidance and insight to the direction of the project.

Bibliography

[1] P. van de Witte, P.J. Dijkstra, J.W.A. van den Berg, and J. Feijen. Phase separation processes in polymer solutions in relation to membrane formation. *Journal of Membrane Science*, 117(1):1 – 31, 1996.

[2] Gregory R. Guillen, Yinjin Pan, Minghua Li, and Eric M. V. Hoek. Preparation and characterization of membranes formed by nonsolvent induced phase separation: A review. *Industrial & Engineering Chemistry Research*, 50:3798–3817, 2011.

[3] Agnieszka K. Holda and Ivo F.J. Vankelecom. Understanding and guiding the phase inversion process for synthesis of solvent resistant nanofiltration membranes. *Journal of Applied Polymer Science*, 132:42130, 2015.

[4] Jeong F. Kim, Ji Hoon Kim, Young Moo Lee, and Enrico Drioli. Thermally induced phase separation and electrospinning methods for emerging membrane applications: A review. *AIChE Journal*, 62:461–490, 2016.

[5] X.L. Wang, H.J. Qian, L.J. Chen, Z.Y. Lu, and A. Lin. Dissipative particle dynamics simulation on the polymer membrane formation by immersion precipitation. *J. Memb. Sci.*, 311:251–258, 2008.

[6] Y.D. He, Y.H. Tang, and X.L. Wang. Dissipative particle dynamics simulation on the membrane formation of polymer-diluent system via thermally induced phase separation. *J. Memb. Sci.*, 368:78–85, 2011.

[7] Y.H. Tang, Y.D. He, and X.L. Wang. Effect of adding a second diluent on the membrane formation of polymer/diluent system via thermally induced phase separation: Dissipative particle dynamics simulation and its experimental verification. *J. Memb. Sci.*, 409-410:164–172, 2012.

[8] Y.H. Tang, Y.D. He, and X.L. Wang. Three-dimensional analysis of membrane formation via thermally induced phase separation by dissipative particle dynamics simulation. *J. Membr. Sci.*, 437:40–48, 2013.

[9] Y.H. Tang, Y.D. He, and X.L. Wang. Investigation on the membrane formation process of polymer-diluent system via thermally induced phase separation accompanied with mass transfer across the interface. dissipative particle dynamics simulation and its experimental verification. *J. Memb. Sci.*, 474:196–206, 2015.

[10] Y.H. Tang, H.H. Lin, T.Y. Liu, H. Matsuyama, and X.L. Wang. Multiscale simulation on the membrane formation process via thermally induced phase separation accompanied with heat transfer. *J. Memb. Sci*, 515:258–267, 2016.

[11] Han han Lin, Yuan-Hui Tang, Hideto Matsuyama, and Xiao-Lin Wang. Dissipative particle dynamics simulation on the membrane formation of polymer-solvent system via

nonsolvent induced phase separation. *Journal of Membrane Science*, 548:288–297, 2018.

[12] G.T. Caneba and D.S. Soong. Polymer membrane formation through the thermal-inversion process. 2. mathematical modeling of membrane structure formation. *Macromolecules*, 18:2545–2555, 1985.

[13] B.F. Barton and A.J. McHugh. Kinetics of thermally induced phase separation in ternary polymer solutions. i. modeling of phase separation dynamics. *Journal of Polymer Science Part B: Polymer Physics*, 37:1449–1460, 1999.

[14] B.F Barton and A.J McHugh. Modeling the dynamics of membrane structure formation in quenched polymer solutions. *Journal of Membrane Science*, 166:119 – 125, 2000.

[15] G.R. Fernandes, J.C. Pinto, and R Nobrega. Modeling and simulation of the phase-inversion process during membrane preparation. *J. Appl. Polym. Sci.*, 82:3036–3051, 2001.

[16] B. Zhou and A.C. Powell. Phase field simulations of early stage structure formation during imersion precipitation of polymeric membranes in 2d and 3d. *J. Membr. Sci.*, 268:150–164, 2006.

[17] M. Hopp-Hirschler and U. Nieken. Modeling of pore formation in phase inversion processes: model and numerical results. *J. Memb. Sci.*, 564:820–831, 2018.

[18] D.R. Tree, K.T. Delaney, H.D. Ceniceros, T. Iwama, and G.H. Fredrickson. A multi-fluid model for microstructure formation in polymer membranes. *Soft Matter*, 13:3013–3030, 2017.

[19] D.R. Tree, T. Iwama, K.T. Delaney, J. Lee, and G.H. Fredrickson. Marangoni flows during nonsolvent induced phase separation. *ACS Macro Lett.*, 7:582–586, 2018.

[20] DR Tree, LF Dos Santos, CB Wilson, TR Scott, JU Garcia, and GH Fredrickson. Mass-transfer driven spinodal decomposition in a ternary polymer solution. *Soft Matter*, 15: 4614–4628, 2019.

[21] Y. Mino, T. Ishigami, Y. Kagawa, and H. Matsuyama. Three-dimensional phase-field simulations of membrane porous structure formation by thermally induced phase separation in polymer solutions. *J. Membr. Sci*, 483:104–111, 2015.

[22] M. Rosario Cervellere, Yuan-Hui Tang, Xianghong Qian, David M. Ford, and Paul C. Millett. Mesoscopic simulations of thermally-induced phase separation in pvdf/dpc solution. *Journal of Membrane Science*, 577:266–273, 2019.

[23] A. Akthakul, C.E. Scott, A.M. Mayes, and A.J. Wagner. Lattice boltzmann simulations of asymmetric membrane formation by immersion precipitation. *J. Membr. Sci.*, 249: 213–226, 2005.

[24] Yanbiao Gan, Aiguo Xu, Guangcai Zhang, Yingjun Li, and Hua Li. Phase separation in thermal systems: A lattice goltzmann study and morphoogical characterization. *Phys. Rev. E.*, 84:046715, 2011.

[25] Yanbiao Gan, Aiguo Xu, Guangcai Zhang, Ping Zhang, and Yingjun Li. Lattice boltzmann study of thermal phase separation: Effects of heat conduction, viscosity and prandlt number. *A Letter J. Exp. the Frontiers of Phys.*, 97:44002, 2012.

[26] Yanbiao Gan, Aiguo Xu, Guangcai Zhang, and Sauro Succi. Discrete boltzmann modeling of multiphase flows: hydrodynamic and thermodynamic non-equilibrium effects. *Soft Matter*, 11:5336–5345, 2015.

[27] X. He, C. Chen, Z. Jiang, and Y. Su. Computer simulation of formation of polymeric ultrafiltration membrane via immersion precipitation. *J. Memb. Sci.*, 371:108–116, 2011.

[28] J.W. Cahn and J.E. Hilliard. Free energy of a nonuniform system. i. interfacial free energy. *J. Chem. Phys.*, 20:256–267, 1958.

[29] P. Flory. *Principles of Polymer Chemistry.* Cornell University Press, 1971.

[30] Gabriel Tkacik and Leos Zeman. Component mobility analysis in the membrane forming system water/n-methyl-2-pyrrolidone/polyethersulfone. *Journal of Membrane Science*, 31:273–288, 1987.

[31] Leos Zeman and Gabriel Tkacik. Thermodynamic analysis of a membrane-forming system water/n-methyl-2-pyrrolodone/polyethersulfone. *Journal of Membrane Science*, 36:119–140, 1988.

[32] Li Xu and Feng Qui. Simultaneous determination of three floryehuggins interaction parameters in polymer/solvent/nonsolvent systems by viscosity and cloud point measurements. *Polymer*, 55:6795–6802, 2014.

[33] G.D.J. Phillies. Universal scaling equation fro self-diffusion by macromolecules in solution. *Macromolecules*, 19:2367–2376, 1986.

[34] G.D.J. Phillies. The hydrodynamic scaling model for polymer self-diffusion. *J. Phys. Chem*, 93:5029–5039, 1989.

[35] F.W. Althena, J. Smid, J.W.A. Van den Berg, J.G. Wijmans, and C.A. Smoulders. Diiffusion of solvent form a cast cellulose acetate solution during the formation of skinned membranes. *Polymer*, 26(10):1531–1538, September 1985.

[36] J.Y. Kim, Y.D. Kim, T. Kanamori, H.W. Lee, and S.C. Kim. Vitrification phenomena in polysulfone/nmp/water system. *J. Appl. Polym. Sci.*, 71:431–438, 1999.

[37] Jeong Rim Hwang, Seong-Hoe Koo, Jong-Ho Kim, Akon Higuchi, and Tae-Moon Tak. Effects of casting solution composition on performance of poly(ether sulfone) membrane.

J. Appl. Polym. Sci., 60:1343–1348, 1996.

[38] Seong Hyun Yoo, Jong Hak Kim, Jae Young Jho, Jongok Won, and Yong Soo Kang. Influence of the addition of pvp on the morphology of asymmetric polyimide phase inversion membranes: efect of pvp molecular weight. *Journal of Membrane Science*, 236:203–207, 2004.

[39] M. Amirilargani, E. Saljoughi, T. Mohammadi, and M.R. Moghbeli. Effects of coagulation bath temperature and polvinylpyrrolidone content on flat sheet asymmetric polyethersulfone membranes. *Polymer Engineering and Science*, pages 885–893, 2010.

Chapter 6

Conclusion

This dissertation contains two published articles which explore two different membrane formation techniques, TIPS and NIPS. The membrane formation process is highly complex with interconnected casting parameters that lead to a wide variety of pore morphologies. The aim of the models presented here are to bring insight and discovery into how different processing conditions affect membrane morphology and to provide deeper understanding of the physical processes behind membrane formation. The articles included are:

CHAPTER 4: M. Rosario Cervellere, Yuan-Hui Tang, Xianghong Qian, David M. Ford, and Paul C. Millett. Mesoscopic simulations of thermally-induced phase separation in PVDF/DPC solutions. *Journal of Membrane Science*. 266-273, 2019.

CHAPTER 5: M. Rosario Cervellere, Xianghong Qian, David M. Ford, Christina Carbrello, Sal Giglia, and Paul C. Millett. Phase-field modeling of non-solvent induced phase separation (NIPS) for PES/NMP/Water with comparison to experiments. *Journal of Membrane Science*. 118799, 2021.

6.1 Conclusions

The work in this dissertation provide two computational models for simulating membrane formation via TIPS and NIPS which can be adapted do different membrane forming systems. Using the Phase-Field method with the Cahn-Hilliard equation and Flory-Huggins free energy of mixing the models describe the rapid diffusive nature of the membrane formation process and capture various morphologies that occur under various processing conditions. The conclusions of this work are listed below:

1. A phase field model was used to investigate the TIPS process for the system PVDF/DPC. Isotropic and anisotropic quenches were conducted and the resulting membrane morphologies were analyzed for the average pore size and interconnectivity.

2. Isotropic thermal quenches showed that the rate of cooling and polymer concentration has a significant effect on pore size and morphology. Faster quench rates decreased the average pore throughout domain. Increasing the polymer concentration also decreased overall pore size and led to a change in morphology from bicontinuous to cellular when $\bar{\phi}_p$ was increased from 0.20 - 0.25.

3. Anisotropic thermal quenches were used to evaluate the effect that coagulation bath temperature, thermal conductivity, and polymer concenration had on pore size and morphology. A dense pore region formed near the contact surface with the coagulation bath and the thickness of the pore region was also analyzed.

4. Increasing coagulation bath temperature created larger pores throughout the domain while decreasing the thickness of the dense pore region. Increasing thermal conductivity allowed for more rapid heat transfer throughout the domain and decreased the overall pore size while increasing the thickness of the dense pore region. Increasing polymer concentration had a similar effect that isotropic quenches had in that it decreases the overall pore size. Increasing polymer concentration also increased the thickness of the dense pore region.

5. A phase-field model was used to investigate the NIPS process for the membrane forming system PES/NMP/Water. The coagulation bath and polymer concentration were varied to quantify their effect on resulting anisotropic pore structures. Two dimensional and three dimensional simulations were used to capture the cross section and top surface pore morphology. NMP was added to the coagulation bath for comparison to handcast membranes under similar casting conditions.

6. Two dimensional simulations exhibited anisotropic pore size with smaller pores toward the surface in contact with the coagulation bath where a dense pore region forms. Adding NMP to the coagulation bath decreases the thickness of this dense pore region and increases the pore size throughout the entire domain.

7. Three dimensional simulations show the effect adding NMP has with pore size and NMP is added to the coagualation bath the pore size of the top surface increases.

8. Increasing polymer concentration for both two- and three- dimensional simulations decreased overall pore size.

9. Comparisons of the simulations with the experiments show similar trends when adding NMP to the coagulation bath. Adding NMP to the coagulation bath decreases the thickness of the dense pore region, increases the pore size on the top surface and throughout the membrane. The model was not able to capture the small size of pores in the dense pore region of the handcast membranes, however, it's possible to resolve this issue by reducing the grid size and time step size.

6.2 Future Work

Areas of future work for this research are primarily focused on NIPS. The inclusion of additives in polymer solutions is a common technique for commercial filter production. More research needs to be done in order to understand how these additives change thermodynamic and kinetic properties during phase inversion. This could be achieved either with experimental research or by using other modeling techniques such as molecular and dissipative particle dynamics. Future research could also combine TIPS and NIPS to create a more accurate model. Additives can affect thermodynamic behavior of polymer solutions and this is a common practice in industry. Future research could also look into how various process conditions affect co-casting membrane filters. Co-casting is when two polymer films are cast on top of each-other in order to create a more tailored membrane. A combination of all of these methods would create a highly accurate and useful tool for predicting morphologies and advancing the rate at which discovery is made when creating new membrane filters.

Appendix

Analysis scripts for Paper 1 and 2

This section contains the scripts written in C++ used to analyze the VTK outputs from the simulations in Chapters 4 and 5.

Pore Size Algorithm 3D

This algorithm opens each sequential VTK output file and then calculates the two dimensional pore size for each layer and the average pore size for the entire domain. To only calculate the pore size for the final simulation output enter use the value 1 for the number of outputs. This script was used to analyze all simulations in Chapter 4 and the three-dimensional simulations in Chapter 5.

```cpp
#include <iostream>
#include <fstream>
#include <sstream>
#include <cmath>
#include <algorithm>
#include <vector>
#include <ctime>
#include <string>
using namespace std;


int main( int argc, char *argv[])
{
   //start timer
   clock_t begin = clock();
   // check input arguments
   if (argc != 7){      // this value must be 1 more than number of inputs
      cout << "\n Incomplete input. Please provide the following in order:\n";
      cout << " 1. number of outputs \n" << " 2. nstep \n" << " 3. x dimension\n";
      cout << " 4. y dimension\n 5. z dimension\n 6. interface value\n\n";
      return 0;
   }
   //-------------------------------------
   // initialize variables
   //-------------------------------------
   string infilename;
   string outfilename;
```

```
string outfilename2;
stringstream ss;
stringstream ss2;
stringstream ss3;
int ssTemp = 0;
int snapshots = atoi(argv[1]);
int simLength = atoi(argv[2]);
int x = atoi(argv[3]);
int y = atoi(argv[4]);
int z = atoi(argv[5]);
double trans = atof(argv[6]);
fstream infile;
ofstream outfile;
ofstream outfile2;
int spacing = simLength/snapshots;
int maxII = simLength/spacing;
// for 1D scan (y and z)
int interface = 0;
double start = 0.0;
double testStart = 0.0;
double testPoint = 0.0;
double solSpace = 0.0;
double solInter = 0.0;
// for 2D average (y and z planes at depth x)
double solInterAvg = 0.0;
double solInterRun = 0.0;
double diff = 0.0;
double diff2 = 0.0;
double sumDiff2 = 0.0;
double diff3D = 0.0;
double diff3D_2 = 0.0;
double sumDiff3D_2 = 0.0;
double solAvg3D = 0.0;
// ---------------------------------------
//  Allocate Arrays
// ---------------------------------------
// vtkoutput
double ***array3D;
array3D = new double**[x];
for(int i = 0; i < x; i++) {
   array3D[i] = new double*[y];
   for(int j = 0; j < y; j++) {
           array3D[i][j] = new double[z];
     }
}
// average interfacial distance of
// x & y scans for each z depth
```

```
double* b = NULL;
b = new double [z];
for (int k=0; k<z; k++){
   b[k] = 0;
}
// array for (solvent spacing)/(2*interfaces)
// for x & y scan scan
double* D = NULL;
D = new double [z];
for (int k=0; k<2*x; k++){
   D[k] = 0;
}
// array for (solvent spacing)/(2*interfaces)
// for x scan
double* c = NULL;
c = new double [y];
for (int j=0; j<y; j++){
   c[j] = 0;
}
// arrays for standard deviation 2D and 3D
double *stdDev = NULL;
stdDev = new double [y];
for (int j=0; j<y; j++){
   stdDev[j] = 0;
}
double *stdDev3D = NULL;
stdDev3D = new double [z];
for (int k=0; k<z; k++){
   stdDev3D[k] = 0;
}
// --------------------------------------
// open output file for 3D average
// --------------------------------------
ss3 << simLength << "cubicAvg" << ".csv";
outfilename2 = ss3.str();
outfile2.open(outfilename2);
outfile2 << "nstep,poreSpace,+-\n";
// -------------------------------------
// opening files ...
// -------------------------------------
for (int ii = 0; ii < (maxII + 2); ii++) {
   // first c_0.vtk file
   if (ii == 0) {
      ssTemp = ii;
      ss << "c_" << ssTemp << ".vtk";
      infilename = ss.str();
   }
```

```
// second c_1.vtk file
else if (ii == 1) {
   ssTemp = 1;
   ss << "c_" << ii << ".vtk";
   infilename = ss.str();
}
// sequential c_####.vtk files
else {
   ssTemp = (ii - 1)*spacing;
   ss << "c_" << ssTemp << ".vtk";
   infilename = ss.str();
   }
// open inputfile
infile.open(infilename);
if (!infile.is_open()) {
   cout << infilename << endl;
   cout << endl << "Failed to open file: " << infilename << endl;
   //return 0;
}
// -----------------------------------------------
// open output file for 2D scan
// -----------------------------------------------
ss2 << "poreSpacing_" << ssTemp << ".csv";
outfilename=ss2.str();
outfile.open(outfilename);
outfile << "Depth," << ssTemp << "poreSpacing," << "+-" << endl;
// ----------------------------------------------
// start reading input file (vtk header)
// ----------------------------------------------
int header = 0;
string tmp;
while ( header < 12 && infile.good()) {
   getline(infile, tmp);
   header++;
}
// ------------------------------------------------
// loop over 3D space. Count solvent rich deomains
// and count interfaces. calculate average distance
// between droplets
//-------------------------------------------------
// read input file into array3D
for (int k = 0; k < z; k++) {
   for (int j = 0; j < y; j++){
      for (int i= 0; i < x; i++){
         infile >> array3D[i][j][k];
      }
   }
```

```
}
// ------------------------------------------
// scan through array3D
// ------------------------------------------
for (int i = 0; i < x; i++) {
   // --------------------
   // scan along z
   // --------------------
   for (int j = 0; j < y; j++){
      for (int k= 0; k < z; k++){
         testStart = array3D[i][j][k];
         if (i == 0 && testStart < 0.1) solSpace++;
         if (i == x-1) testPoint = array3D[i][j][0];
         else testPoint = array3D[i][j][k+1];
         if (testPoint < 0.1) solSpace++;
         if (testStart < trans && testPoint > trans) interface++;
         else if (testStart > trans && testPoint < trans) interface++;
         testStart = testPoint;
      } // end k
      if (interface == 0 && solSpace == 0) solInter = 0;
      else if (interface == 0 && solSpace != 0) interface = 2;
      // calculate distance between bubbles
      else solInter = solSpace / (0.5*interface);
      // store distance in array
      D[j] = solInter;
      solInterRun += solInter;
      solSpace = 0;
      interface = 0;
      solInter = 0.0;
   } // end j
   // -------------------
   // scan along y
   // -------------------
   for (int aa = 0; aa < z; aa++){
      for (int bb= 0; bb < y; bb++){
         testStart = array3D[i][bb][aa];
         if (bb == 0 && testStart < 0.1) solSpace++;
         if (bb == y-1) testPoint = array3D[i][0][aa];
         else testPoint = array3D[i][bb+1][aa];
         if (testPoint < 0.1) solSpace++;
         if (testStart < trans && testPoint > trans) interface++;
         else if (testStart > trans && testPoint < trans) interface++;
         testStart = testPoint;
      } // end bb aka y
      if (interface == 0 && solSpace == 0) solInter = 0;
      else if (interface == 0 && solSpace != 0) interface = 2;
      else solInter = solSpace / (0.5*interface);  // calculate distance
```

```
}
            between bubbles
        D[y+aa] = solInter;           // store distance in array
        solInterRun += D[aa+y];
        solSpace = 0;
        interface = 0;
        solInter = 0.0;
      } // end aa aka z
      solInterAvg = solInterRun/(z+y);
      b[i] = solInterAvg;
      // standard deviation
      for (int zz = 0; zz<z+y; zz++) {
        diff = b[i] - D[zz];
        diff2 = diff*diff;
        sumDiff2 += diff2;
      }
      // stdDev
      if (b[i] == 0) stdDev[i] = 0;
      else stdDev[i] = sqrt(sumDiff2/(z+y)-1);
      outfile << i << "," << b[i] << "," << stdDev[i] << endl;
      sumDiff2 = 0.0;
      solInterRun = 0.0;
    } // end i
    // --------------------------------
    // average over 3D domain
    // --------------------------------
    double sumB = 0.0;
    double avg3D = 0.0;
    for (int i=0; i<x; i++){
      sumB += b[i];
    }
    avg3D = sumB / x;
    // standard deviation
    for (int zz = 0; zz<x; zz++) {
      diff3D = avg3D - b[zz];
      diff3D_2 = diff3D*diff3D;
      sumDiff3D_2 += diff3D_2;
    }
    if (sumB == 0) stdDev[ii] = 0;
    else stdDev3D[ii] = sqrt(sumDiff3D_2/(x-1));
    outfile2 << ii << ","<< avg3D << "," << stdDev3D[ii] << endl;
    infile.close();
    outfile.close();
    ss.str("");
    ss2.str("");
    ss3.str("");
    sumDiff3D_2 = 0.0;
  }
```

```
    // close average 3D
    outfile2.close();
    // de-allocate arrays
    for (int i = 0; i < x; i++) {
        for (int j = 0; j < y; j++) {
            delete[] array3D[i][j];
        }
    }
    delete[] array3D;
    array3D = NULL;
    delete [] b;
    b = NULL;
    delete [] stdDev;
    stdDev = NULL;
    delete[] stdDev3D;
    stdDev3D = NULL;
    delete[] D;
    D = NULL;
    // stop clock
    clock_t end = clock();
    double elapsed_secs = double(end - begin) / CLOCKS_PER_SEC;
    cout << endl << "Time :  " << elapsed_secs << endl;
}
```

Pore size algorithm 2D-only

This algorithm only calculates the two dimensional pore size along the depth of a simulation for a single VTK output. This simplified version of the algorithm in Section 6.2 was used to calculate the pore sizes for the two-dimensional simulations in Chapter 5.

```
#include <iostream>
#include <fstream>
#include <sstream>
#include <cmath>
#include <algorithm>
#include <vector>
#include <ctime>
#include <string>
using namespace std;

int main( int argc, char *argv[])
{
    //start timer
    clock_t begin = clock();
    // check input arguments
```

```
if (argc != 6){       // this value must be 1 more than number of inputs
   cout << "\n This script opens the vtk file for the step provided and \n";
   cout << " and computes the pore size along the y direction for step in x: \n
      \n";
   cout << "\n Incomplete input. Please provide the following in order:\n";
   cout << " 1. nstep \n" << " 2. x dimension\n";
   cout << " 3. y dimension\n 4. z dimension\n 5. interface value\n\n";
   return 0;
}
//------------------------------------
// initialize variables
//------------------------------------
string infilename;
string outfilename;
string outfilename2;
stringstream ss;
stringstream ss2;
stringstream ss3;
int ssTemp = 0;
int simLength = atoi(argv[1]);
int x = atoi(argv[2]); // for floats, use atof(argv[])
int y = atoi(argv[3]);
int z = atoi(argv[4]);
double trans = atof(argv[5]);
fstream infile;
ofstream outfile;
// for 1D scan (x and y)
int interface = 0;
double start = 0.0;
double testStart = 0.0;
double testPoint = 0.0;
double solSpace = 0.0;
double solInter = 0.0;
// for 2D average (x and y planes at depth z)
double solInterAvg = 0.0;
double solInterRun = 0.0;
double diff = 0.0;
double diff2 = 0.0;

// ---------------------------------------
//  Allocate Arrays
// ---------------------------------------
// vtkoutput
double ***array3D;
array3D = new double**[x];
for(int i = 0; i < x; i++) {
   array3D[i] = new double*[y];
```

```
  for(int j = 0; j < y; j++) {
        array3D[i][j] = new double[z];
     }
}
// array for (solvent spacing)/(2*interfaces)
// for y scan
double* c = NULL;
c = new double [x];
for (int i=0; i<x; i++){
  c[i] = 0;
}

// ------------------------------------
// open inputfile
// ------------------------------------
ss << "c_" << simLength << ".vtk";
  infilename = ss.str();
infile.open(infilename);
if (!infile.is_open()) {
  cout << endl << "Failed to open file: " << infilename << endl;
  return 0;
}
// ---------------------------------------------
// open output file for 2D scan
// ---------------------------------------------
ss2 << "poreSpacing_" << simLength << ".csv";
outfilename=ss2.str();
outfile.open(outfilename);
outfile << "Depth," << "poreSpacing," << endl;// "+-" << endl;
// --------------------------------------------
// start reading input file (vtk header)
// --------------------------------------------
int header = 0;
string tmp;
while ( header < 12 && infile.good()) {
  getline(infile, tmp);
  header++;
}
// ----------------------------------------------
// loop over 3D space. Count solvent rich domains
// and count interfaces. calculate average distance
// between droplets
//-----------------------------------------------
// read input file into array3D
for (int k = 0; k < z; k++) {
  for (int j = 0; j < y; j++){
    for (int i= 0; i < x; i++){
```

```
        infile >> array3D[i][j][k];
      }
    }
  }
  // scan through array2D
  for (int i = 0; i < x; i++) {
    // -------------------
    // scan along y
    // -------------------
    for (int j= 0; j < y; j++){
      testStart = array3D[i][j][k];
      if (j == 0 && testStart < 0.1) solSpace++;
      if (j == y-1) testPoint = array3D[i][0][k];
      else testPoint = array3D[i][j+1][k];
      if (testPoint < 0.1) solSpace++;
      if (testStart < trans && testPoint > trans) interface++;
      else if (testStart > trans && testPoint < trans) interface++;
      testStart = testPoint;
    } // y
    // handleing polymer only
    if (interface == 0 && solSpace == 0) solInter = 0;
    // handeling only pore space
    else if (interface == 0 && solSpace != 0) interface = 2;
    // calculate distance between bubbles
    else solInter = solSpace / (0.5*interface);
    c[i] = solInter; // store distance in array
    solSpace = 0;
    interface = 0;
    solInter = 0.0;
    outfile << i << "," << c[i] << /*"," <<*/ endl;
  } // end i
  // ----------------------------------
  // close files and reset variables
  // ----------------------------------
  infile.close();
  outfile.close();
  ss.str("");
  ss2.str("");
  ss3.str("");
  // de-allocate arrays
  for (int i = 0; i < x; i++) {
    for (int j = 0; j < y; j++) {
      delete[] array3D[i][j];
    }
  }
  delete[] array3D;
  array3D = NULL;
```

```
    delete [] c;
    c = NULL;
    // stop clock
    clock_t end = clock();
    double elapsed_secs = double(end - begin) / CLOCKS_PER_SEC;
    cout << endl << "Time :  " << elapsed_secs << endl;
}
```

Hoshen-Kopelmann Algorithm

This Hoshen-Kopelmann algorithm calculates the interconnectivity of pores and was used in Chapter 4. A more detailed explanation can be found in Section 3.4.

```
#include <iostream>
#include <fstream>
#include <cmath>
#include <algorithm>
#include <vector>
#include <ctime>
#include <string>
using namespace std;

//new_cluster will be called when no relevant neighbors contain non-zero values
//It will increase max_label, assign that value in arr, and add it to labels

int new_cluster(int &max_label, vector<int> &labels)
{
    max_label++;
    labels[max_label] = max_label;
    return max_label;
}
//Find will return the proper value for the equivalence class described by the
//input argument
int find(int x, vector<int> &labels)
{
    int y = x;
    while (labels[y] != y)
        y = labels[y];

    while (labels[x] != x)
    {
        int z = labels[x];
        labels[x] = y;
        x = z;
    }
```

```
  return y;
}
//make_union recognizes labeling conflicts and unifies clusters into the same
//equivalence class
void make_union(int x, int y, vector<int> &labels)
{

  labels[find(y, labels)] = find(x, labels);
  labels[y] = labels[x];
}
// ---------------------------------------------------
// Begin HK algorithm
// ---------------------------------------------------
int main( int argc, char *argv[])
{
  clock_t begin = clock(); //start timer
  if (argc != 8 ){
    cout << " Incomplete input. Please provide the following in order: " << endl;
    cout << " 1. Input file" << endl << " 2. Output file" << endl << " 3. nx";
    cout << endl << " 4. ny" << endl << " 5. nz" << " 6. c range (less/greater)";
    cout << endl << " 7. c range threshold (e.g. 0.3)";
    return 0;
  }

  // ------------------------------------------
  // Initialize variables
  // ------------------------------------------

  string InFilename;
  string OutFilename;
  InFilename = argv[1];
  OutFilename = argv[2];
  fstream infile;
  ofstream outfile;
  double point;
  int x = atoi(argv[3]);
  int y = atoi(argv[4]);
  int z = atoi(argv[5]);
  string range = argv[6];
  double cutoff = atof(argv[7]);
  int n = x*y*z;
  double phase = 0;
  double continuity = 0;
  int max_clusters = n / 2;
  vector<int> labels(max_clusters, 0); //vector to store equivalence classes
  vector< vector<int> > labels2(labels.size(), vector<int>(2,0)); //vector of
     vectors for re-work of labels
```

```
vector<int> neighbors(6,0);
vector<int> neighbors2(6,0);
int c_count = 0;
int noise = 0;
int max_neighbor = 0;
int min_neighbor = 0;
int neighbor_count = 0;
int max_label = 0;
int k = 0;        //current height
int i = 0;        //row number
int j = 0;        //column number
int up = 0;       //initialize neighbor variable
int down = 0;     //initialize neighbor variable
int left = 0;     //initialize neighbor variable
int right = 0;    //initialize neighbor variable
int top = 0;      //initialize neighbor variable
int bottom = 0;   //initialize neighbor variable
// initialize array
int*** arr = new(nothrow) int**[z];
for ( k = 0; k < z; k++) {
   arr[k] = new(nothrow) int*[y];
   for ( i = 0; i < y; i++) {
      arr[k][i] = new(nothrow) int[x];
   }
}
// open file
infile.open(InFilename);
outfile.open(OutFilename);
if (!infile.is_open()){
   cout << endl << "Failed to open file" << endl;
   return 0;
}
// go through header of VTK file to get to data
int header = 0;
string tmp;
while ( header < 12 && infile.good()) {
   getline(infile, tmp);
   outfile << tmp << endl;
   header++;
}
// -----------------------------------
// begin scan
// -----------------------------------
for (int k = 0; k < z; k++) {
   for (int i = 0; i < y; i++){
      for (int j = 0; j < x; j++){
         infile >> point;
```

```
        if (range=="less") {
          if (point < cutoff) {
            arr[k][i][j] = 1;
          }
          else
            arr[k][i][j] = 0;
        }
        else if ( range=="greater") {
          if (point > cutoff) {
            arr[k][i][j] = 1;
          }
          else
            arr[k][i][j] = 0;
        }
      }
    }
  }
  outfile.close();
  infile.close();

  // ------------------------------------------------------------
  // Initial pass scan of arr, placing first labels
  // and creating equivalence classes contained
  // in vector labels
  // ------------------------------------------------------------
   for (int k = 0; k < z; k++)
  {
    for (i = 0; i < y; i++)
    {
      for (j = 0; j < x; j++)
      {
        if (arr[k][i][j] > 0)
        {
          if (i == 0)
            down = 0;
          else
            down = labels[arr[k][i - 1][j]];
          if (j == 0)
            left = 0;
          else
            left = labels[arr[k][i][j-1]];
          if (i == y - 1)
            up = labels[arr[k][0][j]];
          else
            up = 0;
          if (j == x - 1)
            right = labels[arr[k][i][0]];
```

```
else
   right = 0;
if (k == 0)
   bottom = 0;
else
   bottom = labels[arr[k-1][i][j]];
if (k == z - 1)
   top = labels[arr[0][i][j]];
else
   top = 0;
// Populate neighbors vector and count relevant neighbors
neighbors = {up, down, left, right, top, bottom};
neighbor_count = 0;
for (int m = 0; m < neighbors.size(); m++){
   if (neighbors[m] > 0)
      neighbor_count++;
}
max_neighbor = *max_element(neighbors.begin(),neighbors.end());
sort(neighbors.begin(), neighbors.end());
bool min_check = false;
for (int m = 0; m < neighbors.size(); m++){
   if (min_check == false && neighbors[m] != 0){
      min_neighbor = neighbors[m];
      min_check = true;
   }
}
// Create new cluster if no neighbors are relevant
if (neighbor_count == 0){
   arr[k][i][j] = new_cluster(max_label, labels);
}
// Assign value of relevant neighbor if only one exists
else if (neighbor_count == 1){
   arr[k][i][j] = max_neighbor;
}
// Determine smallest neighbor cluster value and merge relevant
    neighbors into that cluster
else {
   for (int m = 0; m < neighbors.size(); m++){
      neighbors2[m] = neighbors[m];
      if (neighbors2[m] == 0){
         neighbors2[m] = 1000000;
      }
   }
   int min_m = 0;
   int min_n2 = *min_element(neighbors2.begin(), neighbors2.end());
   for (int m = 0; m < neighbors2.size(); m++){
      if (neighbors2[m] == 1000000){
```

```
                neighbors2[m] = 0;
              }
              if (neighbors2[m] == min_n2){
                min_m = m;
              }
            }
            for (int m = 0; m < neighbors2.size(); m++){
              arr[k][i][j] = neighbors[min_m];
              if (neighbors[m] != 0){
                if (neighbors[min_m] <= neighbors[m] && neighbors[min_m] <=
                    min_neighbor)
                  make_union(neighbors[min_m], neighbors[m], labels);
              }
            }
          }
        }
      }
    }
  }
  // Assure creation of proper equivalence classes
  for (int m = max_clusters - 1; m > 0; m--){
    vector<int> eqClass(1,m);
    if (labels[m] != 0 && labels[m] != m){

      int m2 = m;
      while (labels[m2] != m2){
        eqClass.push_back(labels[m2]);
        m2 = labels[labels[m2]];
      }
      eqClass.push_back(m2);
      int val = *min_element(eqClass.begin(), eqClass.end());
      for ( int p = eqClass.size() - 1; p >= 0; p--){
        labels[eqClass[p]] = val;
      }
    }
    eqClass = vector<int>();
  }
  for (int m = 0; m < max_clusters; m++){
    if (labels[m] != 0){
      labels[m] = labels[labels[m]];
    }
  }
  //Re-work labels to be consecutive
  for (int i = 0; i < labels.size(); i++){
    labels2[i][1] = labels[i];
    labels2[i][0] = i;
  }
```

```
sort(labels2.begin(), labels2.end(), [](const std::vector< int >& a, const
    std::vector< int >& b){ return a[1] < b[1]; });
int old_label = 0;
int new_label = 0;
 bool check = false;
 for (int i = 2; i < labels.size(); i++){
   if(check == true && labels2[i][1] != old_label){
      check = false;
   }
   if(check == true){
      labels2[i][1] = new_label;
   }
   if (labels2[i][1] > labels2[i-1][1] && check == false){
      old_label = labels2[i][1];
      new_label = labels2[i-1][1] + 1;
      labels2[i][1] = new_label;
      check = true;
   }
 }
sort(labels2.begin(), labels2.end(), [](const std::vector< int >& a, const
    std::vector< int >& b){ return a[0] < b[0]; });
for (int i = 0; i < labels.size(); i++){
   labels[i] = labels2[i][1];
}
int new_max = *max_element(labels.begin(), labels.end());
vector<double> sizes(new_max +1,0);
// Assign newly ordered labels to arr
for (k = 0; k < z; k++) {
   for (i = 0; i < y; i++) {
      for (j = 0; j < x; j++) {
         if (arr[k][i][j] > 0){
            arr[k][i][j] = labels[arr[k][i][j]];
            sizes[arr[k][i][j]] += 1;
         }
      }
   }
}
// ----------------------------------------------------
// write output
// ----------------------------------------------------
outfile.open(OutFilename, ios::app);
for (k = 0; k < z; k++) {
   for (i = 0; i < y; i++) {
      for (j = 0; j < x; j++) {
         outfile << arr[k][i][j] << endl;
      }
```

```
    }
  }
  outfile.close();
  sizes[0] = 0;
  for (int k = 1; k < sizes.size(); k++){
    if (sizes[k] == 1){
      sizes.erase(sizes.begin()+k);
      noise += 1;
    }
  }
  for (int k = 1; k < sizes.size(); k++){
    phase += sizes[k];
  }
  for (int k = 1; k < sizes.size(); k++){
      sizes[k] = sizes[k]/phase;
  }
  double max_size = *max_element(sizes.begin(), sizes.end());
  new_max -= noise;
  cout << new_max << " " << max_size << endl;
  // -----------------------------------------
  // free variables
  // -----------------------------------------
  labels = vector<int>();
  labels2 = vector<vector<int> >();
   neighbors = vector<int>();
  neighbors2 = vector<int>();
   sizes = vector<double>();
  for (k = 0; k < z; k++) {
    for (i = 0; i < y; i++) {
      delete[] arr[k][i];
    }
  }
  delete[] arr;
  // -----------------------------
  // show elapsed time
  // -----------------------------
  clock_t end = clock();
  double elapsed_secs = double(end - begin) / CLOCKS_PER_SEC;
  cout << endl << elapsed_secs << endl;
  return 0;
}
```